Particle Methods for Change Detection, System Identification, and Control

CHRISTOPHE ANDRIEU, ARNAUD DOUCET, SUMEETPAL S. SINGH, AND VLADISLAV B. TADIĆ

Invited Paper

Particle methods are a set of powerful and versatile simulation-based methods to perform optimal state estimation in nonlinear non-Gaussian state-space models. The ability to compute the optimal filter is central to solving important problems in areas such as change detection, parameter estimation, and control. Much recent work has been done in these areas. The objective of this paper is to provide a detailed overview of them.

Keywords**—***Change detection, control, optimal filtering, parameter estimation, sequential Monte Carlo, state-space models, stochastic approximation.*

I. Introduction

Optimal filtering for nonlinear non-Gaussian state-space models has numerous applications in signal processing and related areas such as finance, robotics, telecommunications, etc. However, except for simple models such as linear Gaussian state-space models, optimal filters do not typically admit a closed-form expression. Standard approximation schemes can be unreliable (e.g., the extended Kalman filter) or difficult to implement (e.g., deterministic integration methods). Sequential Monte Carlo (SMC) methods, also known as particle methods, are simulation-based approximations of the posterior distributions of interest. These flexible and powerful methods have become very popular over the last few years and the diversity of applications is increasing, e.g., weather forecasting, bioinformatics, etc.

In this paper, we will not discuss standard applications of SMC methods to state estimation. Instead, we will focus on the application of the SMC principle to solve nontrivial problems in the following areas.

- *Model validation/Change detection*: The question one tries to answer in this context is that of determining how well a given model fits the data or how to detect an abrupt change in some parameter values.
- *Parameter estimation/System identification*: Here, given a model with unknown static parameters, the aim is to design both offline and online algorithms to estimate these parameters.
- *Control*: Assuming that it is possible to control the state-space model, the aim is to find a control policy that minimizes a given cost function of the states, observations and controls. Such problems arise in target tracking applications. For example, in sensor management, one must select at each time step a sensor, from a collection of sensors, that measures some aspect of the state of the target being tracked. The aim is to select sensors to optimize the tracking performance.

This is a tutorial paper and contains only a few original developments. It mostly attempts to summarize and present as concisely as possible the main ideas of the recent works published on the subjects mentioned above. We have kept the level of notation and technicality to a minimum and, instead of focusing on a rigourous mathematical development, we have tried to focus on methodological aspects.

The paper is organized as follows. In Section II, we briefly present a generic SMC method for optimal filtering and likelihood evaluation in nonlinear non-Gaussian state-space models. Methods to compute the derivative of the optimal filter and the likelihood function with respect to some parameters are reviewed. Section III presents some algorithms

Manuscript received April 26, 2003; revised December 10, 2003. The work of A. Doucet was supported by an EPSRC grant.

C. Andrieu is with the Department of Mathematics, Bristol University, BS8 1TW Bristol, U.K. (e-mail: C.Andrieu@bristol.ac.uk).

A. Doucet the Department of Engineering, Cambridge University, CB2 1PZ Cambridge, U.K. (e-mail: ad2@eng.cam.ac.uk).

S. S. Singh was with the Electrical Engineering Department, Melbourne University, Melbourne, Vic. 3010, Australia. He is now with the Department of Engineering, Cambridge University, CB2 1PZ Cambridge, U.K. (e-mail: ssss@ee.mu.oz.au).

V. B. Tadić is with the Department of Automatic Control, Sheffield University, Mappin Street, S1 3JD Sheffield, U.K. (e-mail: v.tadic@sheffield.ac.uk).

Digital Object Identifier 10.1109/JPROC.2003.823142

for model validation and change detection. Section IV reviews the various methods to perform offline and online parameter estimation while Section V shows how some control problems can be addressed using particle methods. Finally, we discuss a few open problems in Section VI. A detailed list of references is provided.

II. State-Space Models and Sequential Monte Carlo Methods

A. State-Space Models

Let $\{X_n\}_{n\geq 1}$ and $\{Y_n\}_{n\geq 0}$ be $\mathcal{X}$- and $\mathcal{Y}$-valued stochastic processes defined on a measurable space $(\Omega, \mathcal{F})$. These stochastic processes depend on a parameter $\theta \in \Theta$ where Θ will be assumed to be an open subset of $\mathbb{R}^{n_\theta}$. The process $\{X_n\}_{n\geq 1}$ is an unobserved (hidden) Markov process with initial density μ; i.e., $X_0 \sim \mu$, and Markov transition density $f_\theta(x' \,|\, x)$; i.e.,

$$X_{n+1} \,|\, X_n = x \sim f_\theta(\cdot \,|\, x). \tag{1}$$

Indirect measurements of $\{X_n\}$ are available through the observed process $\{Y_n\}_{n\geq 0}$. More precisely it is assumed that the observations are conditionally independent given $\{X_n\}_{n\geq 1}$ and that their marginal conditional densities are of the form $g_\theta(y \,|\, x)$; i.e.,

$$Y_n \,|\, X_n = x \sim g_\theta(\cdot \,|\, x). \tag{2}$$

All densities are defined with respect to appropriate dominating measures on their spaces which we denote commonly by $d(\,\cdot\,)$; e.g., Lebesgue or counting measures. The interpretation of θ depends on the application under study and will be made clear in the subsequent sections.

This class of models includes many nonlinear and non-Gaussian time series models such as

$$X_{n+1} = \varphi_\theta(X_n, V_{n+1}), \quad Y_n = \psi_\theta(X_n, W_n)$$

where $\{V_n\}_{n\geq 0}$ and $\{W_n\}_{n\geq 0}$ are mutually independent sequences of independent random variables and functions $\varphi_\theta, \psi_\theta$ determine the evolution of the state and observation processes.

B. Optimal Filter

Here and in the following subsections, we will assume that the parameter θ is *known*.

Optimal filtering consists of estimating recursively in time the sequence of posterior densities $\{p_\theta(x_{0:n} \,|\, Y_{0:n})\}_{n\geq 0}$ which summarizes all the information about the system states $X_{0:n}$ as given by the collection of observations $Y_{0:n}$. We have adopted the following notation: for any sequence $\{z_k\}$ and random process $\{Z_k\}$, we define $z_{i:j} = (z_i, z_{i+1}, \ldots, z_j)$ and $Z_{i:j} = (Z_i, Z_{i+1}, \ldots, Z_j)$.

Using Bayes' rule, one can check that the sequence of joint posterior densities $\{p_\theta(x_{0:n} \,|\, Y_{0:n})\}$ satisfies the following recursion:

$$\begin{aligned} p_\theta(x_{0:n} \,|\, Y_{0:n}) \\ &= \frac{g_\theta(Y_n \,|\, x_n) f_\theta(x_n \,|\, x_{n-1}) p_\theta(x_{0:n-1} \,|\, Y_{0:n-1})}{p_\theta(Y_n \,|\, Y_{0:n-1})} \end{aligned} \tag{3}$$

where $p_\theta(Y_n \,|\, Y_{0:n-1})$ is given by

$$\begin{aligned} p_\theta(Y_n \,|\, Y_{0:n-1}) = \int \cdots \int g_\theta(Y_n \,|\, x_n) f_\theta(x_n \,|\, x_{n-1}) \\ \times p_\theta(x_{0:n-1} \,|\, Y_{0:n-1}) dx_{0:n}. \end{aligned} \tag{4}$$

Marginalizing over $x_{0:n-1}$, one obtains the following standard prediction-update recursion for the filtering density $p_\theta(x_n \,|\, Y_{0:n})$. The prediction step to obtain the one-step-ahead prediction density $p_\theta(x_n \,|\, Y_{0:n-1})$ is given by

$$p_\theta(x_n \,|\, Y_{0:n-1}) = \int f_\theta(x_n \,|\, x_{n-1}) p_\theta(x_{n-1} \,|\, Y_{0:n-1}) dx_{n-1}. \tag{5}$$

Then, when the new observation Y_n becomes available and by using Bayes' formula, one obtains the new filtering density

$$p_\theta(x_n \,|\, Y_{0:n}) = \frac{g_\theta(Y_n \,|\, x_n) p_\theta(x_n \,|\, Y_{0:n-1})}{p_\theta(Y_n \,|\, Y_{0:n-1})}. \tag{6}$$

Using (5), the density $p_\theta(Y_n \,|\, Y_{0:n-1})$ given by (3) can be rewritten as

$$p_\theta(Y_n \,|\, Y_{0:n-1}) = \int g_\theta(Y_n \,|\, x_n) p_\theta(x_n \,|\, Y_{0:n-1}) dx_n. \tag{7}$$

We now introduce formal alterations of these formulas, which will prove to be useful when developing SMC methods. Let us define a so-called *importance density* $q_\theta(x_n \,|\, Y_n, x_{n-1})$. The relevance and signification of this density will become clearer when discussing SMC methods later. It is easy to check that one can also rewrite (3) as

$$\begin{aligned} p_\theta(x_{0:n} \,|\, Y_{0:n}) \\ = \frac{\alpha_\theta(x_{n-1:n}, Y_n) q_\theta(x_n \,|\, Y_n, x_{n-1}) p_\theta(x_{0:n-1} \,|\, Y_{0:n-1})}{p_\theta(Y_n \,|\, Y_{0:n-1})} \end{aligned} \tag{8}$$

where

$$\alpha_\theta(x_{n-1:n}, Y_n) = \frac{g_\theta(Y_n \,|\, x_n) f_\theta(x_n \,|\, x_{n-1})}{q_\theta(x_n \,|\, Y_n, x_{n-1})} \tag{9}$$

is the so-called *importance weight*. The relationship (8) is valid as long as

$$\begin{aligned} g_\theta(Y_n \,|\, x_n) f_\theta(x_n \,|\, x_{n-1}) p_\theta(x_{n-1} \,|\, Y_{0:n-1}) > 0 \\ \Rightarrow q_\theta(x_n \,|\, Y_n, x_{n-1}) > 0. \end{aligned}$$

Marginalizing over $x_{0:n-1}$, one obtains (10) and (11) (see bottom of next page), the recursion for the filtering density $p_\theta(x_n \,|\, Y_{0:n})$. The expressions (8), (10), and (11) are, respectively, (3), (5)–(6), and (7) when $q_\theta(x_n \,|\, Y_n, x_{n-1}) = f_\theta(x_n \,|\, x_{n-1})$.

C. The Likelihood

The (marginal) likelihood of $Y_{0:n}$ is given by

$$p_\theta(Y_{0:n}) = \int \cdots \int \underbrace{\mu(x_0)\prod_{k=1}^{n} f_\theta(x_k \,|\, x_{k-1})}_{\text{prior}} \underbrace{\prod_{k=0}^{n} g_\theta(Y_k \,|\, x_k)}_{\text{likelihood}} dx_{0:n}. \tag{12}$$

It also admits the following recursive form:

$$\begin{aligned} p_\theta(Y_{0:n}) &= p_\theta(Y_n \,|\, Y_{0:n-1}) p_\theta(Y_{0:n-1}) \\ &= \prod_{k=0}^{n} p_\theta(Y_k \,|\, Y_{0:k-1}) \end{aligned} \tag{13}$$

with the convention $p_\theta(Y_0 \,|\, Y_{-1}) \triangleq \int g_\theta(Y_0 \,|\, x_0)\mu(x_0)dx_0$. In practice, one uses the log-likelihood which is numerically better behaved and satisfies

$$l_\theta(Y_{0:n}) \triangleq \log p_\theta(Y_{0:n}) = \sum_{k=0}^{n} \log p_\theta(Y_k \,|\, Y_{0:k-1}). \tag{14}$$

Except in a few simple cases including finite state-space hidden Markov models (HMMs) and linear Gaussian state-space models, it is impossible to compute the optimal filter and the log-likelihood/likelihood in closed-form and numerical approximation schemes are required.

D. Optimal Filter and Log-Likelihood Sensitivities

We will see in the following sections that many algorithms require one to be able to compute both the gradients of the optimal filter and of the likelihood with respect to the parameter θ. For simplicity, we will denote by $\nabla_m \Psi_\theta$ the derivative of a function Ψ of θ with respect to the mth component of $\theta, \nabla\Psi_\theta = (\nabla_1\Psi_\theta, \ldots, \nabla_{n_\theta}\Psi_\theta)^{\mathrm{T}}, \nabla_m\Psi_{\theta_n}$ the derivative of a function Ψ with respect to to the mth component of θ evaluated at θ_n and $\nabla\Psi_{\theta_n} = (\nabla_1\Psi_{\theta_n}, \ldots, \nabla_{n_\theta}\Psi_{\theta_n})^{\mathrm{T}}$. From now onwards we will assume that all derivatives written down are well defined.

Taking the derivative of (8), one has (15), shown at the bottom of the page. Note that $\nabla_m p_\theta(x_{0:n} \,|\, Y_{0:n})$ is obviously not a probability density function anymore and actually satisfies $\int \nabla_m p_\theta(x_{0:n} \,|\, Y_{0:n}) dx_{0:n} = 0$, under weak regularity assumptions.

Using (14) one obtains the so-called score function

$$\nabla l_\theta(Y_{0:n}) = \sum_{k=0}^{n} \frac{\nabla p_\theta(Y_k \,|\, Y_{0:k-1})}{p_\theta(Y_k \,|\, Y_{0:k-1})} \tag{16}$$

where using (11)

$$\begin{aligned} &\nabla p_\theta(Y_k \,|\, Y_{0:k-1}) \\ &\quad = \int\int \nabla\alpha_\theta(x_{k-1:k}, Y_k) \\ &\qquad \cdot q_\theta(x_k \,|\, Y_k, x_{k-1}) p_\theta(x_{k-1} \,|\, Y_{0:k-1}) dx_{k-1:k} \\ &\qquad + \int\int \alpha_\theta(x_{k-1:k}, Y_k) \\ &\qquad \cdot \nabla(q_\theta(x_k \,|\, Y_k, x_{k-1}) p_\theta(x_{k-1} \,|\, Y_{0:k-1})) dx_{k-1:k}. \end{aligned} \tag{17}$$

Except in simple cases, it is impossible to compute the gradients of the optimal filter and the log-likelihood function in closed form, and one, therefore, resorts to numerical approximation schemes. In Section II-E below, we detail the use of SMC to approximate the optimal filter while in Section II-F we show how SMC may be used to approximate the gradients of interest.

$$p_\theta(x_n \,|\, Y_{0:n}) = \frac{\int \alpha_\theta(x_{n-1:n}, Y_n) q_\theta(x_n \,|\, Y_n, x_{n-1}) p_\theta(x_{n-1} \,|\, Y_{0:n-1}) dx_{n-1}}{p_\theta(Y_n \,|\, Y_{0:n-1})} \tag{10}$$

and

$$p_\theta(Y_n \,|\, Y_{0:n-1}) = \int\int \alpha_\theta(x_{n-1:n}, Y_n) q_\theta(x_n \,|\, Y_n, x_{n-1}) p_\theta(x_{n-1} \,|\, Y_{0:n-1}) dx_{n-1:n}. \tag{11}$$

$$\begin{aligned} \nabla p_\theta(x_{0:n} \,|\, Y_{0:n}) = &\frac{\nabla\alpha_\theta(x_{n-1:n}, Y_n) \cdot q_\theta(x_n \,|\, Y_n, x_{n-1}) p_\theta(x_{0:n-1} \,|\, Y_{0:n-1})}{p_\theta(Y_n \,|\, Y_{0:n-1})} \\ &+ \frac{\alpha_\theta(x_{n-1:n}, Y_n) \cdot \nabla[q_\theta(x_n \,|\, Y_n, x_{n-1}) p_\theta(x_{0:n-1} \,|\, Y_{0:n-1})]}{p_\theta(Y_n \,|\, Y_{0:n-1})} \\ &- \frac{p_\theta(x_{0:n} \,|\, Y_{0:n})}{p_\theta(Y_n \,|\, Y_{0:n-1})} \\ &\times \left(\int \nabla\alpha_\theta(x_{n-1:n}, Y_n) \cdot q_\theta(x_n \,|\, Y_n, x_{n-1}) p_\theta(x_{0:n-1} \,|\, Y_{0:n-1}) dx_{0:n} \right. \\ &\left. + \int \alpha_\theta(x_{n-1:n}, Y_n) \cdot \nabla[q_\theta(x_n \,|\, Y_n, x_{n-1}) p_\theta(x_{0:n-1} \,|\, Y_{0:n-1})] dx_{0:n} \right). \end{aligned} \tag{15}$$

E. SMC Approximation to the Optimal Filter and Likelihood

1) An SMC Algorithm: We here briefly describe a generic SMC method to approximate the optimal filter based on the sampling resampling approach. More elaborate algorithms are reviewed in [15] and in this special issue.

Assume that at time $n-1$, a collection of $N (N \gg 1)$ random samples named particles $\{\hat{X}_{0:n-1}^{(i)}\}(i = 1, \dots N)$ distributed approximately according to $p_\theta(x_{0:n-1} \,|\, Y_{0:n-1})$ is available. Now at time n, one wishes to produce N particles distributed approximately according to $p_\theta(x_{0:n} \,|\, Y_{0:n})$. An efficient way of doing this consists of setting $\tilde{X}_{0:n-1}^{(i)} = \hat{X}_{0:n-1}^{(i)}$ and then sample $\tilde{X}_n^{(i)} \sim q_\theta(\cdot \,|\, Y_n, \hat{X}_{n-1}^{(i)})$.[1] It follows that the empirical distribution of the particles $\{\tilde{X}_{0:n}^{(i)}\}$ approximates the joint density $p_\theta(x_{0:n-1} \,|\, Y_{0:n-1}) q_\theta(x_n \,|\, Y_n, x_{n-1})$.

By substituting this empirical distribution into (8), one obtains the following approximation of $p_\theta(x_{0:n} \,|\, Y_{0:n})$:

$$\tilde{p}_\theta(x_{0:n} \,|\, Y_{0:n}) = \sum_{i=1}^{N} \alpha_n^{(i)} \delta\left(x_{0:n} - \tilde{X}_{0:n}^{(i)}\right) \tag{18}$$

(where δ represents the delta Dirac mass) i.e., each particle $\tilde{X}_{0:n}^{(i)}$ has now a weight $\alpha_n^{(i)}$ where

$$\alpha_n^{(i)} \propto \alpha_\theta\left(\tilde{X}_{n-1:n}^{(i)}, Y_n\right) \quad \text{and} \quad \sum_{i=1}^{N} \alpha_n^{(i)} = 1.$$

At this stage it is possible to select particles, i.e., particles with small (normalized) weights are discarded and particles with high (normalized) weights are cloned. Many such resampling schemes have been proposed in the literature; see [15]. To sum up, the algorithm proceeds as follows at time n.

```
*Sampling step*
• For $i = 1, \dots, N$, set $\tilde{X}_{0:n-1}^{(i)} = \hat{X}_{0:n-1}^{(i)}$ and sample $\tilde{X}_n^{(i)} \sim q_\theta(\cdot \,|\, Y_n, \hat{X}_{n-1}^{(i)})$.
• For $i = 1, \dots, N$, evaluate the importance weights
    $\alpha_n^{(i)} \propto \alpha_\theta\left(\tilde{X}_{n-1:n}^{(i)}, Y_n\right), \quad \sum_{i=1}^{N} \alpha_n^{(i)} = 1.$
*Resampling step*
• Multiply/Discard particles $\{\tilde{X}_{0:n}^{(i)}\}$ with respect to the high/low weights $\{\alpha_n^{(i)}\}$ to obtain particles $\{\hat{X}_{0:n}^{(i)}\}$.
```

The computational complexity of this algorithm is in $O(N)$. The memory requirements are $O(N(n+1))$ if one stores the whole paths $\{\hat{X}_{0:n}^{(i)}\}$. However, if one is only interested in estimating the marginal density $p_\theta(x_n \,|\, Y_{0:n})$, then the only memory requirements to update the algorithm are $O(2N)$ to store $\{\alpha_n^{(i)}, \tilde{X}_n^{(i)}\}$.

Two possible approximations of the filtering density $p_\theta(x_{0:n} \,|\, Y_{0:n})$ can be performed. Before resampling, one has (18) and after resampling

$$\hat{p}_\theta(x_{0:n} \,|\, Y_{0:n}) = \frac{1}{N} \sum_{i=1}^{N} \delta\left(x_{0:n} - \hat{X}_{0:n}^{(i)}\right). \tag{19}$$

The approximation after resampling is always worse than before resampling as the resampling step introduces additional variance in the estimate. However, resampling is beneficial in the long term, as the algorithm would otherwise collapse.

Particle methods generate at time n an approximation of the joint distribution $p_\theta(x_{0:n} \,|\, Y_{0:n})$ and not only of the marginal filtering distribution $p_\theta(x_n \,|\, Y_{0:n})$. However, only the particle approximation of the marginal distribution $p(x_{n-L+1:n} \,|\, Y_{0:n})$ over a small lag L, say $L \approx 5$, can be used "safely." Indeed as particles are being resampled at each step, the number of distinct trajectories $\{\hat{X}_{0:k}^{(i)}\}$ for a fixed k decreases exponentially quickly as n increases. In practice, one typically focuses on the marginal filtering distribution $p_\theta(x_n \,|\, Y_{0:n})$, so this is not an issue. However, if one is interested in $p_\theta(x_{0:n} \,|\, Y_{0:n})$ or some marginal, say, $p_\theta(x_k \,|\, Y_{0:n}), k < n - L$, then it is practically necessary to rely on alternative smoothing algorithms.

2) Pointwise Particle Approximation of the Likelihood: Based on these approximations of the filtering distributions, it is possible to propose an approximation of the likelihood function. Using (11), one clearly has

$$\hat{p}_\theta(Y_n \,|\, Y_{0:n-1}) = \frac{1}{N} \sum_{i=1}^{N} \alpha_\theta\left(\tilde{X}_{n-1:n}^{(i)}, Y_n\right). \tag{20}$$

Note that if the importance density is chosen equal to the prior f_θ, then (20) becomes

$$\hat{p}_\theta(Y_n \,|\, Y_{0:n-1}) = \frac{1}{N} \sum_{i=1}^{N} g_\theta\left(Y_n \,|\, \tilde{X}_n^{(i)}\right).$$

If the importance density is optimal in terms of minimization of $\mathrm{var}_{q_\theta}[\alpha_\theta(x_{n-1:n}, Y_n) \,|\, x_{n-1}]$, that is

$$q_\theta(x_n \,|\, Y_n, x_{n-1}) = \frac{g_\theta(Y_n \,|\, x_n) f_\theta(x_n \,|\, x_{n-1})}{\int g_\theta(Y_n \,|\, x_n) f_\theta(x_n \,|\, x_{n-1}) dx_n}$$

then (20) becomes

$$\hat{p}_\theta(Y_n | Y_{0:n-1}) = \frac{1}{N} \sum_{i=1}^{N} \int g_\theta(Y_n | x_n) f_\theta\left(x_n | \hat{X}_{n-1}^{(i)}\right) dx_n.$$

If the resampling scheme is unbiased, i.e., the expected number of times a particle is copied in the resampling scheme is equal to its normalized weight, one can show that (20) is an unbiased estimate of $p_\theta(Y_n \,|\, Y_{0:n-1})$. However, the estimate of the log-likelihood is obviously biased

$$\hat{l}_\theta(Y_{0:n}) = \sum_{k=0}^{n} \log \hat{p}_\theta(Y_k \,|\, Y_{0:k-1}). \tag{21}$$

[1] One could actually sample $\tilde{X}_n^{(i)} \sim q_\theta(\cdot \,|\, Y_{0:n}, \hat{X}_{0:n-1}^{(i)})$ but this generalization is not useful for the class of models considered here.

This bias can be reduced by using the following standard correction technique based on a first-order Taylor expansion; i.e., as $N \to \infty$, one typically has $E[\hat{p}_\theta(Y_{0:n})] = p_\theta(Y_{0:n})$ and $\operatorname{var}[\hat{p}_\theta(Y_{0:n})] = (\sigma^2/N)$ where here E and var are an expectation and a variance with respect to the randomness of the algorithm. The constant σ^2 can be easily estimated by, say, $\hat{\sigma}^2$ using the particles. One has

$$E(\hat{l}_\theta(Y_{0:n})) = E(\log(\hat{p}_\theta(Y_{0:n}))) \simeq \log(p_\theta(Y_{0:n})) - \frac{1}{2}\frac{\sigma^2}{N[p_\theta(Y_{0:n})]^2}$$

so a biased-corrected estimate of the log-likelihood is given by

$$\log \widehat{p_\theta(Y_{0:n})} \simeq \hat{l}_\theta(Y_{0:n}) + \frac{1}{2}\frac{\hat{\sigma}^2}{N\exp(2\hat{l}_\theta(Y_{0:n}))}. \tag{22}$$

3) Smooth Particle Method for Likelihood Function Approximation: Assume that one is interested in estimating the log-likelihood function for various values of θ, as required for example to perform maximum-likelihood (ML) parameter estimation or to compute a generalized likelihood ratio (GLR). The true likelihood function is typically a continuous function of θ and the estimate (20) is asymptotically $(N \to \infty)$ consistent. However, for a reasonable number of particles, the variance of the estimates of the log-likelihood might preclude a proper evaluation and maximization of the likelihood surface. An importance sampling method is described in [29] but its computational complexity is in $O(N^2)$ and it is only valid in the neighborhood of a suitably preselected parameter value. We present here a more efficient method recently proposed by Pitt to devise a "smooth" estimate of the likelihood function [45].

The basic idea consists of explicitly rewriting the estimate of the log-likelihood as a deterministic function of the (pseudo)random numbers, say, $\{U_k\}_{k=1,\ldots,n}$ used to sample and resample the particles until time n, i.e.,

$$\hat{l}_n(\theta) = F(\theta, \{U_k\}_{k=1,\ldots,n}). \tag{23}$$

If the mapping $F(\theta, \{U_k\}_{k=1,\ldots,n})$ is continuous in θ given $\{U_k\}_{k=1,\ldots,n}$, then by using the *same random numbers* (i.e., $\{U_k\}_{k=1,\ldots,n}$ fixed) to estimate the likelihood function at all points of interest, one would obtain a "smooth" approximation of the likelihood function. Note that the existence of this mapping excludes the use of rejection sampling or any other procedure using a random number of random variables to generate particles.

Now any standard SMC algorithm in the literature, including the one presented in this section, does not lead to a function $F(\theta, \{U_k\}_{k=1,\ldots,n})$ continuous in θ. This is due to the fact that the resampling operation is not continuous in θ. Indeed it consists of resampling from a discrete point mass function (18) whose weights depend on θ; so if for a value, say, θ_0 one resamples at time n a particle, say, $\tilde{X}_n^{(i)}$ then for a value θ_1 very close to θ_0 one might resample $\tilde{X}_n^{(i-1)}$ or $\tilde{X}_n^{(i+1)}$ instead. Pitt has proposed a "smooth" version of the resampling operation in the scalar state case, i.e., $\mathcal{X} = \mathbb{R}$. The particles $\{\tilde{X}_n^{(i)}\}$ are reordered, so that $\tilde{X}_n^{(1)} < \tilde{X}_n^{(2)} < \ldots < \tilde{X}_n^{(N)}$ and a continuous piecewise linear approximation of the cumulative distribution function associated with $p_\theta(x_n \mid Y_{0:n})$ instead of the piecewise constant (discontinuous) cumulative distribution function associated with (18) is proposed. One resamples the particles from this new cumulative distribution function, therefore "smoothing" the resampling operation; see [45] for details. The computational complexity of this algorithm is in $O(N \log N)$ as one needs to reorder particles, which is still reasonable for most applications. When $\mathcal{X} = \mathbb{R}^{n_x}$ with $n_x > 1$, it seems unfortunately difficult to avoid a computational complexity of order $O(N^2)$.

F. SMC Approximation to the Optimal Filter and Likelihood Gradients

In this section, we focus on the approximation of the filter and likelihood derivatives. We present a particle method recently introduced to address this problem [19]. An alternative method has been proposed independently in [9] for a specific continuous-time model. Other approaches are also discussed.

1) Importance Sampling: At time n, after resampling, one has the approximation (19) of the filtering density. We propose to approximate here the derivative $\nabla_m p_\theta(x_{0:n} \mid Y_{0:n})$ using the same collection of particles $\{\hat{X}_{0:n}^{(i)}\}$ we used to approximate the filter (19), i.e.,

$$\widehat{\nabla_m p_\theta}(x_{0:n} \mid Y_{0:n}) = \sum_{i=1}^{N} \beta_{n,m}^{(i)} \delta\left(x_{0:n} - \hat{X}_{0:n}^{(i)}\right) \tag{24}$$

where the coefficients $\beta_{n,m}^{(i)}$ can be negative or positive and do not sum to one as $\nabla_m p_\theta(x_{0:n} \mid Y_{0:n})$ is not a probability density function. Approximating $\widehat{\nabla_m p_\theta}(x_{0:n} \mid Y_{0:n})$ by (24) corresponds to an importance sampling strategy with importance sampling density $p_\theta(x_{0:n} \mid Y_{0:n})$. The rationale for this approximation is due to the following property. Under weak assumptions, generalizing the arguments in [9], if for any set $A, \int_A p_\theta(x_{0:n-1} \mid Y_{0:n-1})dx_{0:n-1} = 0 \Rightarrow \int_A \nabla_m p_\theta(x_{0:n-1} \mid Y_{0:n-1})dx_{0:n-1} = 0$ then for any set $B, \int_B p_\theta(x_{0:n} \mid Y_{0:n})dx_{0:n} = 0 \Rightarrow \int_B \nabla_m p_\theta(x_{0:n} \mid Y_{0:n})dx_{0:n} = 0$.

To derive a particle method to implement (15), one writes

$$\begin{aligned}
&\nabla(q_\theta(x_n \mid Y_n, x_{n-1})p_\theta(x_{0:n-1} \mid Y_{0:n-1})) \\
&= \nabla q_\theta(x_n \mid Y_n, x_{n-1}) \cdot p_\theta(x_{n-1} \mid Y_{0:n-1}) \\
&\quad + q_\theta(x_n \mid Y_n, x_{n-1}) \cdot \nabla p_\theta(x_{0:n-1} \mid Y_{0:n-1}) \\
&= \frac{\nabla q_\theta(x_n \mid Y_n, x_{n-1})}{q_\theta(x_n \mid Y_n, x_{n-1})} \\
&\quad \cdot q_\theta(x_n \mid Y_n, x_{n-1})p_\theta(x_{0:n-1} \mid Y_{0:n-1}) \\
&\quad + q_\theta(x_n \mid Y_n, x_{n-1}) \cdot \nabla p_\theta(x_{0:n-1} \mid Y_{0:n-1}).
\end{aligned}$$

It follows that the algorithm proceeds as follows at time n. We use the notation $\mathbb{I}_A(z) = 1$ if $z \in A$ and 0 otherwise.

Sampling step

- For $i = 1, \ldots, N$, set $\tilde{X}^{(i)}_{0:n-1} = \hat{X}^{(i)}_{0:n-1}$ and sample $\tilde{X}^{(i)}_n \sim q_\theta(\cdot \,|\, Y_n, \tilde{X}^{(i)}_{n-1})$.
- For $i = 1, \ldots, N$, compute the weights

$$\hat{\beta}^{(i)}_{n,m} = \frac{1}{N} \frac{\nabla_m q_\theta \left(\tilde{X}^{(i)}_n \,|\, Y_n, \tilde{X}^{(i)}_{n-1} \right)}{q_\theta \left(\tilde{X}^{(i)}_n \,|\, Y_n, \tilde{X}^{(i)}_{n-1} \right)} + \beta^{(i)}_{n-1,m}.$$

Updating step

- For $i = 1, \ldots, N$, evaluate the importance weights

$$\alpha^{(i)}_n \propto \alpha_\theta \left(\tilde{X}^{(i)}_{n-1:n}, Y_n \right), \quad \sum_{i=1}^{N} \alpha^{(i)}_n = 1.$$

$$\tilde{\beta}^{(i)}_{n,m} = \frac{\nabla_m \alpha_\theta \left(\tilde{X}^{(i)}_{n-1:n}, Y_n \right) + N \alpha_\theta \left(\tilde{X}^{(i)}_{n-1:n}, Y_n \right) \hat{\beta}^{(i)}_{n,m}}{\sum_{j=1}^{N} \alpha_\theta \left(\tilde{X}^{(j)}_{n-1:n}, Y_n \right)} - \frac{\alpha^{(i)}_n}{\sum_{j=1}^{N} \alpha_\theta \left(\tilde{X}^{(j)}_{n-1:n}, Y_n \right)} \times \left(\sum_{j=1}^{N} \left(\nabla_m \alpha_\theta \left(\tilde{X}^{(j)}_{n-1:n}, Y_n \right) + N \alpha_\theta \left(\tilde{X}^{(j)}_{n-1:n}, Y_n \right) \hat{\beta}^{(j)}_n \right) \right).$$

Resampling step

- Multiply/Discard particles $\{\tilde{X}^{(i)}_{0:n}\}$ with respect to the high/low weights $\alpha^{(i)}_n$ to obtain N particles $\{\hat{X}^{(i)}_{0:n}\}$, i.e., $\hat{X}^{(i)}_{0:n} = \tilde{X}^{(\varphi_n(i))}_{0:n}$ where $\varphi_n(i)$ is determined by the resampling mechanism.
- set

$$\beta^{(i)}_{n,m} = \frac{\tilde{\beta}^{+}_{n,m}}{(\tilde{\beta}/\alpha)^{+}_{n,m}} \frac{\tilde{\beta}^{(\varphi_n(i))}_{n,m}}{\alpha^{(\varphi_n(i))}_n} \mathbb{I}_{\mathbb{R}^+} \left(\tilde{\beta}^{(\varphi_n(i))}_{n,m} \right) + \frac{\tilde{\beta}^{-}_{n,m}}{(\tilde{\beta}/\alpha)^{-}_{n,m} \frac{\tilde{\beta}^{(\varphi_n(i))}_{n,m}}{\alpha^{(\varphi_n(i))}_n} \mathbb{I}^{-}_{\mathbb{R}} \left(\tilde{\beta}^{(\varphi_n(i))}_{n,m} \right)},$$

with

$$\tilde{\beta}^{+}_{n,m} \triangleq \sum_{i=1}^{N} \tilde{\beta}^{(i)}_{n,m} \mathbb{I}_{\mathbb{R}^+} \left(\tilde{\beta}^{(i)}_{n,m} \right),$$

$$(\tilde{\beta}/\alpha)^{+}_{n,m} \triangleq \sum_{i=1}^{N} \tilde{\beta}^{(\varphi_n(i))}_{n,m} / \alpha^{(\varphi_n(i))}_n \mathbb{I}_{\mathbb{R}^+} \left(\tilde{\beta}^{(\varphi_n(i))}_{n,m} \right),$$

$$\tilde{\beta}^{-}_{n,m} \triangleq \sum_{i=1}^{N} \tilde{\beta}^{(i)}_{n,m} \mathbb{I}_{\mathbb{R}^-} \left(\tilde{\beta}^{(i)}_{n,m} \right),$$

and

$$(\tilde{\beta}/\alpha)^{-}_{n,m} \triangleq \sum_{i=1}^{N} \tilde{\beta}^{(\varphi_n(i))}_{n,m} / \alpha^{(\varphi_n(i))}_n \mathbb{I}_{\mathbb{R}^-} \left(\tilde{\beta}^{(\varphi_n(i))}_{n,m} \right).$$

In this algorithm, the derivative $\nabla_m (q_\theta(x_n | Y_n, x_{n-1}) p_\theta(x_{0:n-1} | Y_{0:n-1}))$ is approximated by

$$\sum \hat{\beta}^{(i)}_{n,m} \delta \left(x_{0:n} - \tilde{X}^{(i)}_{0:n} \right) \tag{25}$$

and, before resampling, $\nabla_m p_\theta(x_{0:n} \,|\, Y_{0:n})$ is approximated by

$$\sum \tilde{\beta}^{(i)}_{n,m} \delta \left(x_{0:n} - \tilde{X}^{(i)}_{0:n} \right). \tag{26}$$

Using (17), it follows that one has

$$\widehat{\nabla_m p_\theta}(Y_k \,|\, Y_{0:k-1}) = \frac{1}{N} \sum_{i=1}^{N} \nabla_m \alpha_\theta \left(\tilde{X}^{(i)}_{k-1:k}, Y_k \right) + \sum_{i=1}^{N} \hat{\beta}^{(i)}_{n,m} \alpha_\theta \left(\tilde{X}^{(i)}_{k-1:k}, Y_k \right) \tag{27}$$

and an estimate of the score function (16) is given by (28), shown at the bottom of the page.

2) Alternative Methods: Several alternative approaches to estimate $\nabla p_\theta(x_{0:n} \,|\, Y_{0:n})$ using particle methods are possible. The most natural one consists of using two distinct set of particles for $p_\theta(x_{0:n} \,|\, Y_{0:n})$ and $\nabla p_\theta(x_{0:n} \,|\, Y_{0:n})$.

In this approach, one uses the fact that under regularity assumptions the derivative of a probability density function $v_\theta(z)$ with respect to the mth component of θ can be written as

$$\nabla_m v_\theta(z) = c_{v,\theta,m} (v'_{\theta,m}(z) - v''_{\theta,m}(z))$$

where $c_{\nu,\theta,m} \in \mathbb{R}^+, v'_{\theta,m}(z)$ and $v''_{\theta,m}(z)$ are two probability density functions. There is actually an infinity of such decompositions; the most obvious one is the standard Hahn–Jordan decomposition where $v'_{\theta,m}(z) \propto \max(0, \nabla_m v_\theta(z))$ and $v''_{\theta,m} \propto -\min(0, \nabla_m v_\theta(z))$. However, this might not be the most convenient decomposition and for standard densities there exist more appropriate decompositions where $v'_{\theta,m}$ and $v''_{\theta,m}$ are themselves standard densities [43].

A first approach consists of using such a decomposition for the importance sampling density. One writes

$$\begin{aligned} \nabla_m q_\theta(x_n \,|\, Y_n, x_{n-1}) &= c_{q,\theta,m}(x_{n-1}, Y_n)(q'_{\theta,m}(x_n \,|\, Y_n, x_{n-1}) \\ &\quad - q''_{\theta,m}(x_n \,|\, Y_n, x_{n-1})) \end{aligned} \tag{29}$$

and modifies the sampling step of the algorithm described in the previous subsection. Indeed, one has

$$\begin{aligned} \nabla_m q_\theta(x_n \,|\, Y_n, x_{n-1}) \cdot p_\theta(x_{n-1} \,|\, Y_{0:n-1}) &= c_{q,\theta,m}(x_{n-1}, Y_n)(q'_{\theta,m}(x_n \,|\, Y_n, x_{n-1}) \\ &\quad \times p_\theta(x_{n-1} \,|\, Y_{0:n-1}) \\ &\quad - q''_{\theta,m}(x_n \,|\, Y_n, x_{n-1}) p_\theta(x_{n-1} \,|\, Y_{0:n-1})). \end{aligned}$$

$$\widehat{\nabla_m l_\theta}(Y_{0:n}) = \sum_{k=0}^{n} \frac{\sum_{i=1}^{N} \nabla_m \alpha_\theta \left(\tilde{X}^{(i)}_{k-1:k}, Y_k \right) + N \sum_{i=1}^{N} \hat{\beta}^{(i)}_{n,m} \alpha_\theta \left(\tilde{X}^{(i)}_{k-1:k}, Y_k \right)}{\sum_{i=1}^{N} \alpha_\theta \left(\tilde{X}^{(i)}_{k-1:k}, Y_k \right)}. \tag{28}$$

So by sampling particles from both $q'_{\theta,m}(x_n \mid Y_n, x_{n-1})$ and $q''_{\theta,m}(x_n \mid Y_n, x_{n-1})$, it is possible to obtain an approximation of $\nabla_m q_\theta(x_n \mid Y_n, x_{n-1}) \cdot p_\theta(x_{n-1} \mid Y_{0:n-1})$ instead of using importance sampling. One can then plug this approximation in (15) and use the standard Hahn–Jordan decomposition for $\nabla_m p_\theta(x_{0:n} \mid Y_{0:n})$. This is easily implemented in practice by checking the sign of the weight of each particle; see [20].

A second approach consists of also using such decompositions for $\nabla_m p_\theta(x_{0:n} \mid Y_{0:n})$ and $\nabla_m \alpha_\theta(x_{n-1:n}, Y_n)$,[2] i.e.,

$$\begin{aligned}\nabla_m p_\theta(x_{0:n} \mid Y_{0:n}) &= c_{p,\theta,m}(Y_{0:n})(p'_{\theta,m}(x_{0:n} \mid Y_{0:n}) \\ &\quad - p''_{\theta,m}(x_{0:n} \mid Y_{0:n})) \\ \nabla_m \alpha_\theta(x_{n-1:n}, Y_n) &= c_{\alpha,\theta,m}(x_{n-1:n})(\alpha'_{\theta,m}(x_{n-1:n}, Y_n) \\ &\quad - \alpha''_{\theta,m}(x_{n-1:n}, Y_n)).\end{aligned}$$

Using such decompositions and plugging them in (15), it is trivial to obtain recursions for $c_{p,\theta,m}(Y_{0:n})$, $p'_{\theta,m}(x_{0:n} \mid Y_{0:n})$, and $p''_{\theta,m}(x_{0:n} \mid Y_{0:n})$, which can be implemented using particle methods.

III. Model Validation and Change Detection

The problems of model validation, change/fault detection, and isolation in dynamic models have received considerable attention; see, for example, [6] and [26]. We discuss here a few applications of particle methods to these problems.

A. Model Validation

Assume here the parameter θ is fixed. Model validation is the process of determining how well a given model fits the data. Within a Bayesian framework, models can be compared using posterior model probabilities, but this strategy only provides relative performance indicators and does not tell whether any particular model fits the data well. In this section, we show how particles and frequentist methods can be combined to determine the goodness of fit for any model of the data.

Under the hypothesis that the model is the correct one, it is straightforward to show [47] that the sequence $\{u_k\}_{k=1,\ldots,n}$ with

$$u_k = \Pr(Y_k \le y_k \mid Y_{0:k-1} = y_{0:k-1}, \theta)$$

is a realization of independent identically distributed (i.i.d.) variables uniformly distributed on $[0,1]$ given a realization of the observations $y_{0:n}$. This result holds true for any time series and may be used in statistical tests to determine the adequacy of the model. Computing $\{u_k\}_{k=1,\ldots,n}$ can be generally performed using particle methods. Indeed, one has

$$u_k = \int \Pr(Y_k \le y_k \mid X_k = x_k, \theta) p_\theta(x_k \mid y_{0:k-1}) dx_k \tag{30}$$

and one can obtains a particle approximation of the one-step ahead prediction distribution by using a particle approximation of $p_\theta(x_{k-1} \mid y_{0:k-1})$ given by

$$\sum_{i=1}^{N} \alpha_{k-1}^{(i)} \delta\left(x_{k-1} - \tilde{X}_{k-1}^{(i)}\right)$$

and by sampling $\tilde{X}_k^{(i)} \sim f_\theta(\cdot \mid \tilde{X}_{k-1}^{(i)})$. It yields

$$\hat{u}_k = \sum_{i=1}^{N} \alpha_{k-1}^{(i)} \Phi_\theta\left(\tilde{X}_k^{(i)}, y_k\right)$$

where

$$\Phi_\theta(x, y_k) = \int_{\{y \le y_k\}} g_\theta(y \mid x) dy. \tag{31}$$

If (31) cannot be expressed analytically, then another estimate of (30) can be computed by

$$\tilde{u}_k = \sum_{i=1}^{N} \alpha_{k-1}^{(i)} \mathbb{I}_{(Y^{(i)} \le y_k)}$$

where $Y^{(i)} \sim g_\theta(\cdot \mid \tilde{X}_k^{(i)})$. An importance sampling method could also be used.

The estimates obtained for $\{u_k\}_{k=1,\ldots,n}$ may be used instead of the true values to determine the adequacy of the model. Most of these tests are based on the transformation of the sequence $\{u_k\}_{k=1,\ldots,n}$ into a sequence $\{v_k\}_{k=1,\ldots,n}$ where $v_k = \Psi^{-1}(u_k)$, with Ψ the standard Gaussian cumulative distribution function. Thus, under the hypothesis that the model is correct the $\{v_k\}_{k=1,\ldots,n}$ are i.i.d. distributed according to $\mathcal{N}(0,1)$. One can then employ standard statistical tests such as Bownman–Shenton or Ljung–Box to test the normality and whiteness of $\{v_k\}_{k=1,\ldots,n}$. In [56], this method is applied to speech signals.

B. Change Detection Based on Likelihood Ratios

1) Method: We follow closely here the approach proposed in [38]. Let us first consider a simple change detection problem. Before the time change, we assume that we have a dynamic model defined by (1)–(2) with $\theta = \theta_0$. After the change, for sake of simplicity, we will assume that new model is the dynamic model (1)–(2) with $\theta = \theta_1 (\theta_0 \ne \theta_1)$.[3] Given some observations $Y_{0:n}$, we replace the unknown change time by its ML estimate, i.e.,

$$\hat{j}_n = \arg\max_{1 \le j \le n} p_{\theta_0,\theta_1,j}(Y_{0:n})$$

with

$$\begin{aligned}p_{\theta_0,\theta_1,j}(Y_{0:n}) &\triangleq \prod_{k=0}^{n} p_{\theta_0,\theta_1,j}(Y_k|Y_{0:k-1}) \\ &= \prod_{k=0}^{j-1} p_{\theta_0}(Y_k|Y_{0:k-1}) \prod_{k=j}^{n} p_{\theta_0,\theta_1,j}(Y_k|Y_{0:k-1})\end{aligned}$$

[2] Properly normalized, $\alpha_\theta(x_{n-1:n}, Y_n)$ is a probability density of argument Y_n.

[3] This assumption can be easily relaxed. The new model could be any dynamic model with transition and observation equations of functional forms different from (1)–(2).

where $p_{\theta_0,\theta_1,j}(Y_k \mid Y_{0:k-1})$ denotes the predictive density of Y_k given $Y_{0:k-1}$when one has $\theta = \theta_0$ for the time interval $\{0, 1, \ldots, j-1\}$ and $\theta = \theta_1$ afterwards. One can easily check that

$$\hat{j}_n = \underset{1 \leq j \leq n}{\arg\max}\ S_j^n$$

where S_j^n is the following likelihood ratio (LR):

$$S_j^n \triangleq \sum_{k=j}^{n} \log\left(\frac{p_{\theta_0,\theta_1,j}(Y_k \mid Y_{0:k-1})}{p_{\theta_0}(Y_k \mid Y_{0:k-1})}\right). \tag{32}$$

The change detector can be obtained by

$$g_n = S_{\hat{j}_n}^n = \max_{1 \leq j \leq n} S_j^n \lessgtr \lambda$$

where g_n corresponds to the decision function and λ is a threshold. That is, we decide a change has occurred whenever g_n exceeds λ. The change/fault alarm is set at the time given by

$$t_a = \min\{n : g_n > \lambda\} = \min\left\{n : \max_{1 \leq j \leq n} S_j^n > \lambda\right\}$$

and the ML estimate of change onset time t_0 after a change detection is equal to

$$t_0 = \underset{1 \leq j \leq t_a}{\arg\max}\ S_j^n.$$

Until now, we have considered the case where the model after change is known. In many real-world applications, this is not the case, and either the model parameter θ_1 belongs to a finite set ("simple" case) or a continuous set ("difficult case") Θ_1. In this case, a second maximization is required to compute g_k and one requires the GLR

$$g_n = \max_{1 \leq j \leq n} \sup_{\theta_1 \in \Theta_1} S_j^n(\theta_1)$$

as the LR in (32) is now a function of θ_1.

Experimental results obtained in [38] demonstrate the power of this method. However, contrary to the linear Gaussian case, it seems difficult to make any precise theoretical statement on the optimal properties of such a test in the nonlinear non-Gaussian framework.

2) Particle Implementation: Assume θ_1 is fixed. In this case, the change detection algorithm only requires computing (32). This can be done easily using particle methods using (21). However, to compute the LR S_j^n for a given j one needs two particle filters (change at time j and no change). This means that to compute g_n one requires $n + 2$ particle filters: one particle filter for which $\theta = \theta_0$ between 0 and n and $(n + 1)$ particle filters where $\theta = \theta_0$ for $k < j$ and $\theta = \theta_1$ for $k \geq j$ where $j \in \{1, \ldots, n\}$.

When θ_1 is not fixed but belongs to a finite set Θ_1 of cardinality M, one has to use $1 + Mn$ particle filters. This is computationally very intensive. In [38], the authors propose to work over a time window of fixed width Δ so as to limit the computational complexity, whereas in [5] the author proposes to modify this statistical test so as to keep a computational complexity independent of n. Note that it is actually possible to use a particle version of the two-filter smoothing formula [32] in the spirit of [26, p. 355] to limit the computational complexity to $M + 1$ filters and backward filters.

When Θ_1 is a continuous set, one requires to perform the maximization of $S_j^n(\theta_1)$. This can be performed using the ML parameter estimation procedures detailed in Section IV-B. This leads to a quite complex algorithm.

To avoid this, it is possible to use alternatively a local test method, i.e., testing small deviations from the nominal value θ_0. This test only requires computing the score function (16) at $\theta = \theta_0$, which can be done using (28); see [36].

C. Change Detection Based on Multiple Models

When the number of potential models for the data is finite, i.e., corresponding to the scenario in the previous section when Θ_1 is finite, then an alternative approach to change detection is the so-called multiple models one ([26, ch. 10]). In this case, there is, say, K competing models for the data. Formally, let us introduce a latent (i.e., unobserved) process $\{M_n\}_{n \geq 0}$ taking values in a finite set $\mathcal{M} = \{1, 2, \ldots, K\}$ then, when the latent process satisfies $M_n = m$, the system evolves according to

$$X_{n+1} \mid (X_n = x, M_{n+1} = m) \sim f_m(\cdot \mid x) \tag{33}$$

and

$$Y_n \mid (X_n = x, M_n = m) \sim g_m(\cdot \mid x). \tag{34}$$

If $M_n = M$ is a constant random variable, then one can select the model by computing the (penalized) likelihood[4] for each candidate model and selecting the one maximizing it. However, in most applications, the process $\{M_n\}_{n \geq 0}$ is not constant and follows a prior distribution, say, $p(m_{0:n})$. In this case, any Bayesian inference on $(M_{0:n}, X_{0:n})$ given $Y_{0:n}$ relies on the joint posterior distribution $p(m_{0:n}, x_{0:n} \mid Y_{0:n})$. This distribution satisfies

$$\begin{aligned} p(m_{0:n}, x_{0:n}|Y_{0:n}) \propto\, & g_{m_n}(Y_n|x_n) f_{m_n}(x_n|x_{n-1}) \\ & \times p(m_n|m_{0:n-1}) p(m_{0:n-1}, x_{0:n-1}|Y_{0:n-1}). \end{aligned}$$

By considering the extended state $Z_n = (X_n, M_n)$, one can approximate $p(m_{0:n}, x_{0:n} \mid Y_{0:n})$ using particle methods.[5] This is the approach followed recently, for example, by [12], [22], [34], and [37]. Based on the approximation of this distribution, one can, for example, estimate at any time the posterior probabilities of M_n.

We note that making use of the discrete nature of M_n, it is possible to develop elaborated particle methods. When, conditional upon $M_{1:n}$, the model (33) and (34) is linear Gaussian, then one can use a special particle filter corresponding to a mixture of Kalman filters. This method is based on the fact that in this case

$$p(m_{0:n}, x_{0:n} \mid Y_{0:n}) = p(m_{0:n} \mid Y_{0:n}) p(x_{0:n} \mid Y_{0:n}, m_{0:n})$$

where $p(x_{0:n} \mid Y_{0:n}, m_{0:n})$ is a Gaussian distribution of known statistics and is computed by the Kalman filter. It

[4] One can use the Akaike information criterion, minimum description length, or other criteria.

[5] The fact that $\{M_n\}_{n \geq 0}$ might not be Markovian—i.e., $p(m_n \mid m_{0:n-1}) \neq p(m_n \mid m_{n-1})$; thus, $\{Z_n\}_{n \geq 0}$ is not Markovian—is not a problem. Particle methods still apply in this framework.

follows one can concentrate the particles to the approximation of the lower-dimensional distribution $p(m_{0:n} \mid Y_{0:n})$, thus reducing the variance of the Monte Carlo estimates; see, for example, [14] and [16] for details and [12] for a detailed application. The same idea can be used for dynamic Bayesian networks with a large state space [13]. Finally, in the general nonlinear non-Gaussian case, one can still make use of the discrete nature of M_n to develop efficient algorithms; see [3].

IV. Parameter Estimation

In this section, we consider the model (1)–(2) for which the parameter is not known. We will here denote θ^* the true value of the parameter θ. After reviewing briefly particle based methods that have been proposed in order to solve the problem of the estimation of θ^*, we go on to present both batch and online algorithms design to compute point estimates of θ^*.

A. Filtering Methods

A standard approach followed in the literature consists of setting a prior distribution on the parameter θ and then considering the extended state $Z_n \triangleq (X_n, \theta)$. This has the theoretical advantage of converting the static parameter inference problem into an optimal filtering problem. It is then tempting to use standard particle filtering techniques in order to estimate the series of distributions $\{p(x_n, \theta \mid Y_{0:n})\}$ and, thus, $\{p(\theta \mid Y_{0:n})\}$. However, due to the lack of ergodicity of the process $\{Z_n\}$, such approaches are bound to fail. This fact motivated the development of alternative techniques, which we now review.

1) Degeneracy and Kernel Methods: If one were to apply the generic SMC algorithm described in Section II, the parameter space would only be explored at the initialization of the algorithm as the transition probability of the Markov process $\{Z_n\}_{n\geq 0}$ includes a delta-Dirac mass for the component θ. As a consequence, after a few iterations the marginal posterior distribution of the parameter is typically approximated by a single delta Dirac function, which corresponds to one of the initial values sampled from the prior distribution at time 0. This problem was quickly identified, and in order to limit it, several authors proposed using kernel density estimation methods [25], [39]. More precisely, assume that at time n, after resampling, the following approximation of $p(\theta \mid Y_{0:n})$ is available:

$$\hat{p}(\theta \mid Y_{0:n}) = \frac{1}{N}\sum_{i=1}^{N} \delta\left(\theta - \hat{\theta}_n^{(i)}\right) \tag{35}$$

where many particles $\{\hat{\theta}_n^{(i)}\}$ are actually equal. The idea of kernel methods consists of substituting to the degenerate distribution (35) the kernel approximation

$$\frac{1}{N}\sum_{i=1}^{N} K\left(\theta - \hat{\theta}_n^{(i)}\right) \tag{36}$$

where K is a convolution kernel, e.g., Gaussian. It is then possible to sample from (36) to obtain a new set of particles. The advantage of this approach is that it introduces diversity in the set of particles, but this is at the cost of transforming the fixed parameter into a slowly time-varying one whose dynamics is related to the width of the kernel $K(\cdot)$. Additionally, the choice of the kernel width and its effect on the original problem is not always clear.

Another pragmatic approach consists of explicitly introducing artificial dynamics on the "static" parameter of interest [28], [33]; say, e.g.,

$$\theta_{n+1} = \theta_n + \varepsilon_{n+1}$$

where $\{\varepsilon_n\}_{n\geq}$ is an artificial (small) dynamic noise. Again, as for the kernel method approach the choice of an appropriate variance for the artificial dynamic noise is difficult and the original problem has again been modified.

2) Alternative Particle Methods: **Coupling Particle Methods with MCMC.** To avoid the introduction of an artificial dynamic model, an interesting approach proposed in [23] consists of adding Markov chain Monte Carlo (MCMC) steps so as to add "diversity" among the particles. More precisely, assume that at time n the approximation (35) is available. In order to add diversity, one samples new particles according to an MCMC kernel with invariant distribution $p(x_{0:n}, \theta \mid Y_{0:n})$, i.e.,

$$\left(X_{0:n}^{(i)}, \theta_n^{(i)}\right) \sim K_{\text{MCMC}}\left(\cdot, \cdot \mid \hat{X}_{0:n}^{(i)}, \hat{\theta}_n^{(i)}\right)$$

before moving to the sequential importance sampling step. Contrary to standard applications of MCMC, the kernel does not have to be ergodic. It is actually never ergodic in practice, as ensuring ergodicity would require one to sample an increasing number of variables over time—the algorithm would not be sequential anymore. In practice, one typically samples $\theta_n^{(i)}$ and possibly $X_{n-L+1:n}^{(i)}$ for some integer $L \geq 1$, therefore setting $X_{0:n-L}^{(i)} = \hat{X}_{0:n-L}^{(i)}$. Note that the memory requirements for this method does not increase over time if $p(Y_{0:n} \mid \theta, x_{0:n})$ can be summarized by a set of fixed dimensional sufficient statistics. This combination of MCMC and particle methods has also been adopted by Chopin [10]. Its main advantage over standard kernel approximations is that, whatever the choice of the MCMC kernel used, the target distribution $p(x_{0:n}, \theta \mid Y_{0:n})$ is never modified.

Storvik's Method. Storvik [52] has recently proposed an alternative method. It is based on the following decomposition:

$$\begin{aligned} p(x_{0:n}, \theta \mid Y_{0:n}) &\propto g_\theta(Y_n \mid x_n) f_\theta(x_n \mid x_{n-1}) \\ &\quad \times p(\theta \mid Y_{0:n-1}, x_{0:n-1}) p(x_{0:n-1} \mid Y_{0:n-1}). \end{aligned} \tag{37}$$

Assuming that at time $n-1$ a set of particles $\{(\hat{X}_{0:n-1}^{(i)}, \hat{\theta}_{n-1}^{(i)})\}$ distributed according to $p(x_{0:n-1}, \theta \mid Y_{0:n-1})$ is available, one can sample new samples $(\hat{X}_n^{(i)}, \tilde{\theta}_n^{(i)}) \sim q(\cdot, \cdot \mid Y_{0:n}, \hat{X}_{0:n-1}^{(i)})$. Following (37), the importance weights are given by the equation shown at the bottom of the next page. Note that the algorithm is truly recursive if $p(Y_{0:n} \mid \theta, x_{0:n})$ and $q(x_n, \theta_n \mid Y_{0:n}, x_{0:n-1})$ only depend on $x_{0:n}$ and $Y_{0:n}$ through a set of fixed dimensional sufficient statistics.

Problems With These Methods. As opposed to the methods relying on kernel or artificial dynamics, these two approaches have the advantage of adding diversity to the particles approximating $p(\theta \,|\, Y_{0:n})$ without perturbing the target distribution. However, both algorithms suffer from an accumulation of error over time; see [1] for an example showing the MCMC-based algorithm can even diverge over time. The reason for this is that to approximate $p(\theta \,|\, Y_{0:n})$, one tries to compute

$$p(\theta \,|\, Y_{0:n}) = \int p(\theta \,|\, Y_{0:n}, x_{0:n}) p(x_{0:n} \,|\, Y_{0:n}) dx_{0:n}$$

which is an integral whose dimension is increasing over time. It is impossible to obtain an approximation whose error is not increasing over time by using a fixed number of particles and not rejuvenating the particles associated to $x_{0:n}$ from time 0 to n.

In [46], a pragmatic approach is adopted to limit the degeneracy of the paths $p(x_{0:n} \,|\, Y_{0:n})$. The idea is to make the assumption that

$$p(x_{0:k} \,|\, \theta, Y_{0:n}) \simeq p(x_{0:k} \,|\, \theta, Y_{0:n+j}) \tag{38}$$

for any $j \geq 0$ and $n-k$ "large" enough, that is, $x_{0:k}$ has very little influence on $Y_{n+1:n+j}$. Under this mixing assumption, it is possible during the simulation to "freeze" the paths $x_{0:n}$ from time 0 to k after a few time steps as by assumption new observations do not bring any new information about past state values. Although reasonable when the parameter θ is known, the mixing assumption (38) might be questionable when the static parameter is unknown, as the model is not in this case ergodic.

The problem of estimating sequentially in time the series of distributions $\{p(\theta \,|\, Y_{0:n})\}$ seems to be intractable and we now focus on another type of approach, which aims at producing point estimates of θ^*.

B. Point Estimation Methods

We now present some methods referred to as "point estimation" methods; i.e., we do not aim at estimating the series of posterior distributions $\{p(\theta \,|\, Y_{0:n})\}$, and, therefore, do not require the use of particles in the parameter space. We focus rather on the estimation of θ^* directly, using for example the ML principle. These methods can be used to perform offline/online (penalized) ML parameter estimation and rely either on greedy maximization or deterministic/stochastic gradient techniques. It should, therefore, not be surprising if they are found to be sensitive to initialization and might get trapped in a local maximum.

1) Offline Methods: **Greedy Maximization and Standard Gradient Methods.** Assume one is given a fixed set of data $Y_{0:n}$. To perform ML parameter estimation, one could try to perform a greedy maximization as a function of θ of the likelihood given by (14), or, more exactly, the particle approximation of it. In the latter case, to reduce the variance of the estimate of the likelihood (14), one would require many particles. It is also possible to use the "smooth" estimate of the likelihood introduced by Pitt when $n_x = 1$ [45]—see Section II-C.

To perform ML parameter estimation, one may use a gradient algorithm with a decreasing stepsize sequence $\{\gamma_k\}_{k\geq 0}$ so as to average out the "noise" introduced by the Monte Carlo estimates of the log-likelihood. One selects a stepsize sequence satisfying $\gamma_k \to 0, \sum_{k=0}^{\infty} \gamma_k = \infty, \sum_{k=0}^{\infty} \gamma_k^2 = \infty$; e.g., $\gamma_k = C \cdot k^{-\alpha}, C > 0$, and $\alpha \in ((1/2), 1]$. At iteration $k+1$ of the gradient algorithm, we update θ_{k+1} using

$$\theta_{k+1} = \theta_k + \gamma_k \widehat{\nabla l}_{\theta_k}(Y_{0:n})$$

where$\widehat{\nabla l}_{\theta_k}(Y_{0:n})$ is the particle estimate of the likelihood gradient from time 0 to n at the current parameter value θ_k, as given by (28).

EM-type Algorithms. If the smooth likelihood method is used for an unknown multidimensional parameter θ, it can be difficult to properly scale the parameter increments used to perform the greedy maximization. Similarly, it is difficult to properly scale the components of the gradient. In the context of linear Gaussian state-space models and finite state-space HMMs, a very popular alternative method for parameter estimation is the expectation–maximization (EM) algorithm and its variants. The EM algorithm is a numerically well-behaved gradient method which increases the log-likelihood at each iteration. It proceeds as follows. Given an estimate θ_k of the parameter, then at iteration $k+1$

$$\theta_{k+1} = \arg\max_{\theta \in \Theta} Q(\theta, \theta_k)$$

where

$$Q(\theta, \theta_k) = \int \log p_\theta(x_{0:n}, Y_{0:n}) p_{\theta_k}(x_{0:n} \,|\, Y_{0:n}) dx_{0:n}. \tag{39}$$

In the nonlinear non-Gaussian framework, one cannot compute (39) exactly, and it is necessary to use an approximation. Using particle methods with $\theta = \theta_k$, one can obtain an approximation of the joint density $p_{\theta_k}(x_{0:n} \,|\, Y_{0:n})$ which substituted into (39) yields

$$\hat{Q}(\theta, \theta_k) = \sum_{i=1}^{N} \alpha_n^{(i)} \log p_\theta \left(\tilde{X}_{0:n}^{(i)}, Y_{0:n} \right).$$

This function can be easily maximized when $p_\theta(\tilde{X}_{0:n}^{(i)}, Y_{0:n})$ is in the exponential family. In this case $p_\theta(x_{0:n}, Y_{0:n})$ depends on $(x_{0:n}, Y_{0:n})$ only through a set of sufficient statistics $\varphi(x_{0:n}, Y_{0:n})$ whose dimension is independent of

$$\alpha_n^{(i)} \propto \frac{g_{\tilde{\theta}_n^{(i)}}\left(Y_n \,|\, \tilde{X}_n^{(i)}\right) f_{\tilde{\theta}_n^{(i)}}\left(\tilde{X}_n^{(i)} \,|\, \hat{X}_{n-1}^{(i)}\right) p\left(\tilde{\theta}_n^{(i)} \,|\, Y_{0:n-1}, \hat{X}_{0:n-1}^{(i)}\right)}{q\left(\tilde{X}_n^{(i)}, \tilde{\theta}_n^{(i)} \,|\, Y_{0:n}, \hat{X}_{0:n-1}^{(i)}\right)}.$$

n.[6] This method looks attractive but is actually inefficient. This is because the particle approximation of the joint distribution $p_{\theta_k}(x_{0:n} \,|\, Y_{0:n})$ provided by (18) can be poor as soon as n is large. Consequently the variance of $\hat{Q}(\theta, \theta_k)$ would increase exponentially with n. To obtain samples from $p_{\theta_k}(x_{0:n} \,|\, Y_{0:n})$ using particle methods, one can use the forward filtering backward sampling method presented in [24] and [29]. However, to obtain M samples from $p_{\theta_k}(x_{0:n} \,|\, Y_{0:n})$ this method requires $O(NM)$ operations, which is very expensive. In this offline context, a good alternative to particle methods is MCMC methods.

2) Online Methods: We consider here online algorithms to estimate the true parameter θ^*. All these algorithms rely on a nonincreasing positive stepsize sequence $\{\gamma_n\}_{n \geq 0}$ such that $\sum \gamma_n = \infty$ and $\sum \gamma_n^2 < \infty$; typically one selects $\gamma_n = C \cdot n^{-\alpha}$ where $C > 0$ and $\alpha \in (1/2, 1]$. We point out that the methods described here rely on the same principle: an alternative contrast function to the likelihood is defined, and this contrast function requires the evaluation of integration over fixed spaces that do not depend on time.

Recursive ML. Consider the log-likelihood function $l_\theta(Y_{0:n})$ given by (14). Under regularity assumptions, including the stationarity of the state-space model, one has [54]

$$\frac{1}{n} l_\theta(Y_{0:n}) \underset{n \to \infty}{\rightarrow} l(\theta) \tag{40}$$

where $l(\theta)$ is defined at the bottom of the page. In this expression, $\mathcal{P}(\mathcal{X} \times \mathcal{X})$ is the space of probability distributions on $\mathcal{X} \times \mathcal{X}$ and $\lambda_{\theta,\theta^*}(dy, d\mu)$ is the joint invariant distribution of $\{Y_n, q_\theta(x_n \,|\, Y_n, x_{n-1}) p_\theta(x_{n-1} \,|\, Y_{0:n-1})\}$. We have made explicit the dependence of $\lambda_{\theta,\theta^*}(dy, d\mu)$ on both θ and the true parameter θ^*. Maximizing $l(\theta)$ amounts to minimizing the following Kullback–Leibler information measure given by

$$K(\theta, \theta^*) \overset{\triangle}{=} l(\theta^*) - l(\theta) \geq 0. \tag{41}$$

In order to optimize this cost function, one can suggest a recursive ML (RML) algorithm, based on the following stochastic gradient recursion:

$$\theta_{n+1} = \theta_n + \gamma_n \nabla \log p_{\theta_{1:n}}(Y_n \,|\, Y_{0:n-1})$$

where $\nabla \log p_{\theta_{1:n}}(Y_n \,|\, Y_{0:n-1})$ is similar to $\nabla \log p_\theta(Y_n \,|\, Y_{0:n-1})$ except that the filter and its gradient appearing in this expression are now computed using the parameter θ_k at time k. This is this approach followed in [35] for finite state-space HMMs and in [9] and [19] for general state-space models. In the general state-space case, it is necessary to approximate $\nabla \log p_{\theta_{1:n}}(Y_n \,|\, Y_{0:n-1}) = \nabla p_{\theta_{1:n}}(Y_n \,|\, Y_{0:n-1}) / p_{\theta_{1:n}}(Y_n \,|\, Y_{0:n-1})$ based on the particle approximations of $q_{\theta_n}(x_n \,|\, Y_n, x_{n-1}) p_{\theta_{1:n}}(x_n \,|\, Y_{1:n})$ and $\nabla(q_{\theta_n}(x_n \,|\, Y_n, x_{n-1}) p_{\theta_{1:n}}(x_n \,|\, Y_{1:n-1}))$ (computed using the algorithm described in Section II-F using parameter θ_k at time k) and using (20) and (27). Convergence analysis of this algorithm requires nonstandard stochastic approximation results developed in [53] and geometric ergodicity results developed in [54].

Online EM Algorithm. The method presented here is detailed in [2]. Similarly to the offline case, it can be difficult to properly scale the different components of the gradient when the parameter θ is multidimensional. One could suggest using an online version of the EM algorithm, as it is a numerically well-behaved gradient method. Online EM algorithms have been established for finite state-space HMMs and linear Gaussian state-space models [21]. However, similarly to the offline case, although a direct implementation of the online EM algorithm using particle methods is feasible in the nonlinear non-Gaussian state-space case, it would fail in practice because of an accumulation of errors over time.

To prevent the degeneracy inherent to this approach, one can modify the contrast function to minimize. Instead of considering the maximization of the average log-likelihood function which leads to (41), we consider here the so-called split-data likelihood (SDL) also called quasilikelihood as proposed in [48], [49] for finite state-space HMM. In this approach, the data set is divided in blocks of, say, L data and one maximizes the average of the resulting log-SDL. This leads to an alternative Kullback–Leibler contrast function. It can be shown under regularity assumptions that the set of parameters optimizing this contrast function includes the true parameter. We maximize here the average log-SDL using an online EM algorithm. The crucial point here is that the integrals that are needed with this approach are defined on a fixed space, and means that it is possible to define Monte Carlo estimators that have a finite variance.

First for a given L and any $k \geq 1$ we denote $\mathsf{Y}_k = Y_{(k-1)L:kL-1}$ and $\mathsf{x}_k = x_{(k-1)L:kL-1}$. Assume $\{X_n\}_{n \geq 0}$ defined by (1) is a stationary Markov process with invariant density $\pi_\theta(x)$. This implies that once the process $\{X_n\}_{n \geq 0}$ has reached its stationary regime, then for any k, the vectors $(\mathsf{x}_k, \mathsf{Y}_k)$ are *identically distributed* according to

$$\begin{aligned} \overline{p}_\theta(\mathsf{x}_k, \mathsf{Y}_k) = {} & \pi_\theta(x_{(k-1)L}) g_\theta(Y_{(k-1)L} \,|\, x_{(k-1)L}) \\ & \times \prod_{i=(k-1)L+1}^{kL-1} f_\theta(x_i \,|\, x_{i-1}) g_\theta(Y_i \,|\, x_i). \end{aligned} \tag{42}$$

One can naturally introduce the likelihood (in the stationary regime) of the block of data Y_k

$$\overline{p}_\theta(\mathsf{Y}_k) = \int \overline{p}_\theta(\mathsf{x}_k, \mathsf{Y}_k) d\mathsf{x}_k \tag{43}$$

$$l(\theta) = \int \int_{\mathcal{Y} \times \mathcal{P}(\mathcal{X} \times \mathcal{X})} \log \left(\int \int \alpha_\theta(x, x', y) \mu(x, x') \, dx \, dx' \right) \lambda_{\theta,\theta^*}(dy, d\mu)$$

[6] With $Q(\theta, \theta_m)$ being evaluated through Monte Carlo, one cannot guarantee anymore that the log-likelihood function will increase monotonically.

and denote $\bar{l}_\theta(\mathsf{Y}_k) \triangleq \log \overline{p}_\theta(\mathsf{Y}_k)$ the associated log-likelihood. Now we introduce the following so-called split-data (marginal) likelihood of p blocks of $L (L \geq 1)$ consecutive observations

$$\tilde{p}_\theta(\mathsf{Y}_1, \ldots, \mathsf{Y}_p) \triangleq \prod_{k=1}^{p} \overline{p}_\theta(\mathsf{Y}_k). \tag{44}$$

This split-data likelihood ignores the dependency between adjacent data blocks. The underlying motivation for the introduction of this quantity relies on the following property, which parallels the classical scenario where the observations Y_k's are independent, and the true likelihood is, therefore, a simple product of densities. Under fairly general ergodicity conditions, the average log-SDL of $Y_{0:pL-1}$ satisfies when $p \to \infty$

$$\frac{1}{p}\sum_{k=1}^{p} \bar{l}(\mathsf{Y}_k) \to \tilde{l}(\theta) \tag{45}$$

where

$$\tilde{l}(\theta) = \int \bar{l}_\theta(\mathsf{Y}_1)\overline{p}_{\theta^*}(\mathsf{Y}_1)d\mathsf{Y}_1$$

and $\overline{p}_{\theta^*}(\mathsf{Y}_1)$ is the joint distribution of L consecutive observations under the true parameter θ^* in the stationary regime. Note the difference with the standard RML approach in (40). Now, maximizing $\tilde{l}(\theta)$ is equivalent to minimizing $\tilde{K}(\theta, \theta^*) \triangleq \tilde{l}(\theta^*) - \tilde{l}(\theta) \geq 0$ [48]. $\tilde{K}(\theta, \theta^*)$ is, therefore, a valid alternative contrast function which has the clear advantage of involving integral over fixed spaces only: realistic Monte Carlo algorithms can, therefore, be used in this case. Now in practice there is a tradeoff associated with the choice of L. For small L, the algorithm is typically easier to implement but the convergence might be slow. If L is large, the algorithm will converge faster as it mimics the convergence properties of the RML estimate but this will be at the cost of a more complex algorithm.

Now, in order to find θ^* one can use an online EM algorithm. The standard batch EM algorithm would consist here at iteration k of the evaluation of

$$Q(\theta_{k-1}, \theta) = \int \log \overline{p}_\theta(\mathsf{x}_1, \mathsf{Y}_1)\overline{p}_{\theta_{k-1}}(\mathsf{x}_1 \,|\, \mathsf{Y}_1)d\mathsf{x}_1$$

followed by the maximization

$$\theta_k = \arg\max_{\theta \in \Theta} Q(\theta_{k-1}, \theta).$$

The online version computes the average Q function defined by

$$\tilde{Q}(\theta_{k-1}, \theta) = \int Q(\theta_{k-1}, \theta)\overline{p}_{\theta^*}(\mathsf{y}_1)d\mathsf{y}_1$$

by a Monte Carlo method; we use the observations $\{\mathsf{Y}_n\}$ which are naturally distributed according to $\overline{p}_{\theta^*}(\cdot)$. More precisely, once the block of observations Y_k is available, and given our current estimate $\tilde{Q}_{k-1}$ of $\tilde{Q}$, one computes

$$\begin{aligned}\tilde{Q}_k(\theta_{1:k-1}, \theta) &= (1 - \gamma_k)\tilde{Q}_{k-1}(\theta_{1:k-2}, \theta)\\ &\quad + \gamma_k \int \log \overline{p}_\theta(\mathsf{x}_k, \mathsf{Y}_k)\overline{p}_{\theta_{k-1}}(\mathsf{x}_k \,|\, \mathsf{Y}_k)d\mathsf{x}_k\end{aligned} \tag{46}$$

and update the value of the parameter according to

$$\theta_k = \arg\max_{\theta \in \Theta} \tilde{Q}_k(\theta_{1:k-1}, \theta).$$

Note naturally the dependence of the estimator $\tilde{Q}_{k-1}$ of $\tilde{Q}$ on $\theta_1, \theta_2, \ldots, \theta_{k-1}$. In the case where the integral in the previous equation does not have an analytical expression, one can again use a Monte Carlo method, and integrate with respect to $\overline{p}_\theta(\mathsf{x}_k \,|\, \mathsf{Y}_k)$. More precisely one can run a particle filter to sample from this distribution, at least when L is not large, say, $L < 5$. Otherwise, when L is large, one can use, for example, the forward filtering backward sampling algorithm [24]. In both cases the algorithm remains an online algorithm. We refer the reader to [2] for further implementation issues.

Note that although we have restricted ourselves to nonoverlapping data blocks $\{\mathsf{Y}_k\}_{k \geq 1}$ for clarity, it is possible to apply our framework to overlapping blocks. This enables one to update the parameter estimate at the data rate (e.g., see [48]) and it possesses nice asymptotic properties.

Finally, we would like to point out that it is possible to adapt the ideas developed here in order to perform online parameter estimation in general state-space models using computationally very efficient simulation-based methods which do not rely on particle methods [4]; note that their computational efficiency is balanced by a rate of convergence slower than that of the algorithms presented in this section.

V. Control

The control framework presented in this section is more commonly known as a partially observed Markov decision process (POMDP). We will be presenting POMDPs where the state, observation action spaces are uncountable, which is more difficult than the standard finite state–observation–action (finite) POMDP. We refer the reader to [27] for rigourous treatment of "uncountable" POMDPs and to [8] for an exposition on the subject in general. For finite state-space POMDPs, one can solve finite and infinite horizon problems using value iteration [41]. For uncountable POMDPs, the standard approach consists of discretizing the state, action, and observation space to obtain a finite POMDP and solve the latter using value iteration [27]. The methods that we present in this section are not based on discretization.

A. State-Space Models With Control

We consider here nonlinear non-Gaussian state-space models for which it is possible to apply an $\mathcal{A}$-valued control term A_n at time n. More precisely conditional upon $\{A_n\}_{n \geq 0}$, the process $\{X_n\}_{n \geq 0}$ is a Markov process with $X_0 \sim \mu$ and Markov transition density $f(x' \,|\, x, a)$, i.e.,

$$X_{n+1} \,|\, (X_n = x, A_{n+1} = a) \sim f(\cdot \,|\, x, a). \tag{47}$$

The observations $\{Y_n\}_{n \geq 0}$ are conditionally independent with marginal density $g(y \,|\, x, a)$, i.e.,

$$Y_n \,|\, (X_n = x, A_n = a) \sim g(\cdot \,|\, x, a). \tag{48}$$

In the most general case, the control A_{n+1} at time $n+1$ is a function of all the available information at time n which can be summarized by the optimal filter $p(x_n \,|\, Y_{0:n}, A_{0:n})$. We are interested in both finite horizon and infinite horizon control problems whose detailed descriptions will be given in the forthcoming subsections.

Solving optimal control problems for nonlinear non-Gaussian state-space models is a formidable task. Although the optimal controller is the solution to the Dynamic Programming/Bellman recursion, except in very specific cases, like a linear Gaussian state-space model and a quadratic cost function, there is no analytical solution to this recursion. In nonlinear non-Gaussian state-space models, the *value function* admits as argument a probability distribution, and it seems extremely difficult to come up with any sensible approximation to it. This is why, despite its numerous applications, the literature on applications of particle methods for control of nonlinear non-Gaussian models is extremely limited.

B. Finite Horizon Control Problems

Let us introduce a cost function $c : \mathcal{X} \times \mathcal{A} \rightarrow \mathbb{R}^{+}$.[7] In [42] and [50], the authors propose to address the following problem. At time $k-1$, the sequence $A_{0:k-1}$ has been selected and the observations $Y_{0:k-1}$ have been received. One wants to minimize the function defined as

$$J(A_{k:k+H-1}) = \mathbb{E}_{X_{k-1} \sim p(\cdot \,|\, Y_{0:k-1}, A_{0:k-1})} \times \left[\sum_{j=k}^{k+H-1} c(X_j, A_j) \right] \quad (49)$$

over $A_{k:k+H-1}$ where the expectation is with respect to the joint distribution of both the states and the observations $(X_{k:k+H-1}, Y_{k:k+H-1,})$ given that X_{k-1} is distributed according to $p(x_{k-1} \,|\, Y_{0:k-1}, A_{0:k-1})$. Here H is the control horizon. Each control input takes its values in a finite set $\mathcal{A}$ of cardinality K. It is possible to approximate $J(A_{k:k+H-1})$ numerically using particle methods for the K^H possible values of $A_{k:k+H-1}$ and then select the optimal value. Indeed one has

$$\begin{aligned}
&\mathbb{E}_{X_{k-1} \sim p(\cdot \,|\, Y_{0:k-1}, A_{0:k-1})}[c(X_j, A_j)] \\
&\quad = \int c(x_j, A_j) \left(\prod_{l=k}^{j} g(y_l \,|\, x_l, A_l) f(x_l \,|\, x_{l-1}, A_l) \right) \\
&\qquad \times \; p(x_{k-1} \,|\, Y_{1:k-1}, A_{0:k-1}) dx_{k-1:j} dy_{k:j} \\
&\quad = \int c(x_j, A_j) \left(\prod_{l=k}^{j} f(x_l \,|\, x_{l-1}, A_l) \right) \\
&\qquad \times \; p(x_{k-1} \,|\, Y_{1:k-1}, A_{0:k-1}) dx_{k-1:j}.
\end{aligned}$$

To approximate $J(A_{k:k+H-1})$, one obtains samples from $p(x_j \,|\, Y_{1:k-1}, A_{0:j})$ for $j \geq k$ using the particle approximation (18) and then sampling particles $\tilde{X}_j^{(i)} \sim f(\cdot \,|\, \tilde{X}_{j-1}^{(i)}, A_j)$ for $j > k$. Then, one obtains the following approximation to (49):

$$\hat{J}(A_{k:k+H}) = \sum_{j=k}^{k+H-1} \sum_{i=1}^{N} \alpha_n^{(i)} c\left(\tilde{X}_j^{(i)}, A_j \right).$$

In [18], the following problem arising in sensor management [31] is addressed. At time $k-1$, one wants to minimize over $A_k \in \mathcal{A}$

$$\begin{aligned}
J(A_k) &= \mathbb{E}_{X_{k-1} \sim p(\cdot \,|\, Y_{0:k-1}, A_{0:k-1})} \\
&\quad \times [c(p(x_k \,|\, Y_{0:k-1}, A_{0:k}), p(x_k \,|\, Y_{0:k}, A_{0:k}))] \quad (50)
\end{aligned}$$

where $c : \mathcal{P}(\mathcal{X}) \times \mathcal{P}(\mathcal{X}) \rightarrow \mathbb{R}^{+}$ is a measure of "similarity" between the prediction $p(x_k \,|\, Y_{0:k-1}, A_{0:k})$ and the filter $p(x_k \,|\, Y_{0:k}, A_{0:k})$, i.e., one aims to select a control input A_k maximizing the average information brought by Y_k about X_k. In this case, contrary to (49), the cost function does depend on the unknown future observation Y_k. One has

$$\begin{aligned}
J(A_k) = \int c(p(x_k \,|\, Y_{0:k-1}, y_k, A_{0:k}), p(x_k \,|\, Y_{0:k-1}, A_{0:k})) \\
\times \, p(y_k \,|\, Y_{0:k-1}, A_{0:k}) \, dy_k.
\end{aligned}$$

If $c(\cdot, \cdot)$ is minus the Kullback–Leibler information, then one can easily establish that

$$\begin{aligned}
J(A_k) = \int p(x_k \,|\, Y_{0:k-1}, A_{0:k}) g(y_k \,|\, x_k, A_k) \\
\times \, \log(g(y_k \,|\, x_k, A_k) / p(y_k \,|\, Y_{0:k-1}, A_{0:k})) \, dx_k \, dy_k.
\end{aligned}$$

Again, one can estimate this quantity by sampling from $p(x_k \,|\, Y_{0:k-1}, A_{0:k})$ and $g(y_k \,|\, x_k, A_k)$, for any $A_k \in \mathcal{A}$, using particle methods and then selecting the optimal A_k value. If A_k takes values in a continuous space and $J(A_k)$ is differentiable with respect to A_k, an estimate of the gradient of $J(A_k)$ can be estimated and a stochastic gradient algorithm can be used. This procedure requires the gradient method developed in Section II-F.

In [51], a more complex control problem related to optimal trajectory planning is addressed. At time $k-1$, one wants to minimize over $A_{k:k+H-1}$

$$\begin{aligned}
&J(A_{k:k+H-1}) \\
&\quad = \mathbb{E}_{X_{k-1} \sim p(\cdot \,|\, Y_{0:k-1}, A_{0:k-1})} \\
&\qquad \cdot \Bigg[\Bigg(\varphi(X_{k+H-1}) \; - \int \varphi(x_{k+H-1}) \\
&\qquad\qquad \cdot p(x_{k+H-1} | Y_{0:k+H-1}, A_{0:k+H-1}) dx_{k+H-1} \Bigg)^2 \Bigg] \\
&\quad = \int \Bigg(\varphi(x_{k+H-1}) - \int \varphi(x_{k:k+H-1}) \\
&\qquad\qquad \cdot p(x_{k+H-1} | Y_{0:k-1}, y_{k:k+H-1}, A_{0:k+H-1}) dx_{k+H-1} \Bigg)^2 \\
&\qquad \cdot p(x_{k+H-1}, y_{k:k+H-1} \,|\, Y_{0:k-1}, A_{0:k+H-1}) \\
&\qquad\qquad \cdot dx_{k+H-1} dy_{k:k+H-1}
\end{aligned}$$

where $\varphi : \mathcal{X} \rightarrow \mathbb{R}$, i.e., one wants to find the sequence of control inputs such that the average mean square error between $\varphi(X_{k+H-1})$ and its optimal minimum mean square

[7] One can generalize all the algorithms described below to the case $c : \mathcal{X} \times \mathcal{Y} \times \mathcal{A} \rightarrow \mathbb{R}^{+}$.

estimate is minimum. If $A_{k:k+H-1}$ takes values in a continuous space and $J(A_{k:k+H-1})$ is differentiable with respect to $A_{k:k+H-1}$, then one has

$$\begin{aligned}\nabla_{A_j} J(A_{k:k+H-1}) &= \mathbb{E}_{X_{k-1}\sim p(\cdot\,|\,Y_{0:k-1},A_{0:k-1})}\Bigg\{\Bigg(\varphi(X_{k+H-1}) \\ &- \int \varphi(x_{k:k+H-1}) p(x_{k+H-1}\,|\,Y_{1:k-1}, y_{k:k+H-1}, \\ &\times\ A_{0:k+H-1}) dx_{k+H-1}\Bigg)^2 S_j(Y_j, X_{j-1:j}, A_j)\Bigg\}\end{aligned}$$

where

$$S_j(y, x_{j-1:j}, a) = \frac{\nabla_a g(y\,|\,x_j, a)}{g(y\,|\,x_j, a)} + \frac{\nabla_a f(x_j\,|\,x_{j-1}, a)}{f(x_j\,|\,x_{j-1}, a)}.$$

It is possible to optimize $J(A_{k:k+H-1})$ using a stochastic gradient algorithm. Variance reduction techniques for estimating $\nabla_{A_j} J(A_{k:k+H-1})$ have been developed in [51] to improve the efficiency of the stochastic gradient algorithm.

C. Infinite Horizon Control Problems

In the infinite horizon case, we are interested in selecting a control sequence $\{A_n\}_{n\geq 0}$ to minimize the *infinite horizon discounted cost*

$$J(\mu, \{A_n\}_{n\geq 0}) = \mathbb{E}_{X_0\sim\mu}\left[\sum_{k=0}^{\infty} \gamma^k c(X_k)\right] \qquad (51)$$

where the discount factor $0 < \gamma < 1$, and the expectation is with respect to the joint distribution of both the states and the observations. In [55], a particle method based on Q-learning is proposed to solve for a stationary policy [see (54) below] when $\mathcal{A}$ is a finite set. This method is complex as the Q-factors are functions that admit as arguments probability distributions over $\mathcal{X}$. It is, thus, necessary to perform further approximations. Thrun develops a nearest-neighbor type method to perform quantization in this space [55].

In an (infinite horizon) *average cost* formulation, one has

$$J(\mu, \{A_n\}_{n\geq 0}) \triangleq \lim_{n\to\infty} \frac{1}{n}\sum_{k=1}^{n} \mathbb{E}_{X_0\sim\mu}[c(X_k)] \qquad (52)$$

where the expectation is with respect to the joint distribution of both the states and the observations. We review here two algorithms proposed recently by the authors to solve this problem [17]. These algorithms are not based on any discretization. Instead, we limit the class of admissible policies by defining a family of policies parameterized by a parameter θ. We then search over this space of policies for the optimal one using a stochastic gradient descent (or ascent) algorithm. This approach is known as the *policy gradient* method and hinges on one being able to obtain estimates of the gradient of the performance criterion with respect to θ. As we show, the gradient of the performance criterion involves the filtering distribution, its derivative and integration with respect to it. Thus, in the general state-space setting, one must use a particle method to approximate the performance criterion gradient. This will add a bias to the gradient, but in all the application investigated by us thus far, the bias was observed to be negligible.

More formally, let $\mathcal{P}(\mathcal{E})$ denote the set of probability distributions on a set $\mathcal{E}$. We consider the set of randomized stationary policies $\nu_\theta : \mathcal{P}(\mathcal{X}) \to \mathcal{P}(\mathcal{A})$ of the form

$$A_k \sim \nu_\theta(\cdot\,|\,p(x_{k-1}\,|\,Y_{0:k-1}, A_{0:k-1})) \qquad (53)$$

or deterministic stationary policies $\varphi_\theta : \mathcal{P}(\mathcal{X}) \to \mathcal{A}$ of the form

$$A_k = \varphi_\theta(p(x_{k-1}\,|\,Y_{0:k-1}, A_{0:k-1})). \qquad (54)$$

$\theta \in \Theta \subset \mathbb{R}^{n_\theta}$ is the parameter to be optimized. Note that in selecting A_k, the filtering distribution $p(x_{k-1}\,|\,Y_{0:k-1}, A_{0:k-1})$ is used, which is a function of $(Y_{0:k-1}, A_{0:k-1})$. The density $p(x_{k-1}\,|\,Y_{0:k-1}, A_{0:k-1})$ summarizes the information in the past controls and observations that is relevant for the future control of the system.

When parameterized policies are used, one has $J(\mu, \{A_n\}_{n\geq 0}) = J(\mu, \theta)$. Under an ergodic assumption $J(\mu, \theta) = J(\theta)$; i.e., the cost is independent of the initial distribution μ and is a function of θ only

$$J(\theta) = \lim_{n\to\infty} J_n(\theta) \quad \text{where } J_n(\theta) \triangleq \mathbb{E}_{X_0\sim\mu}[c(X_n)].$$

We propose to optimize θ using a stochastic gradient algorithm. Under additional regularity assumptions, one has

$$\lim_{n\to\infty} \nabla J_n(\theta) = \nabla J(\theta).$$

For a randomized policy (53), one can easily check that an unbiased estimate of the gradient is given by

$$\begin{aligned}\hat{\nabla} J_n(\theta) = \left(\int c(x_n) p(x_n\,|\,Y_{0:n}, A_{0:n}) dx_n\right) \\ \cdot \left(\sum_{k=1}^{n} \frac{\nabla \nu_\theta(A_k\,|\,p(x_{k-1}\,|\,Y_{0:k-1}, A_{0:k-1}))}{\nu_\theta(A_k\,|\,p(x_{k-1}\,|\,Y_{0:k-1}, A_{0:k-1}))}\right). \qquad (55)\end{aligned}$$

In the nonlinear non-Gaussian state-space models, we compute the first term using the particle approximation (18). Note that in practice, a discount factor on the second term on the right side of (55) is required in order to prevent the variance of the gradient estimate from growing. This is at the cost of an added bias. A stochastic gradient algorithm to minimize $J(\theta)$ follows directly; see [17] for details.

For a deterministic policy (54), the notation (54) is inadequate, as one should make explicit the dependency of the filter on θ. In this case, an unbiased estimate of the gradient is given by

$$\begin{aligned}\hat{\nabla} J_n(\theta) = \int c(x_n) \cdot \nabla p_\theta(x_n\,|\,Y_{0:n}, A_{0:n}) dx_n \\ + \left(\int c(x_n) \cdot p_\theta(x_n\,|\,Y_{0:n}, A_{0:n}) dx_n\right) \cdot \nabla l_\theta(Y_{0:n})\end{aligned}$$

where $\nabla l_\theta(Y_{0:n})$ is the score given by (16). In nonlinear non-Gaussian state-space models, we can easily approximate all these terms using (26), (18), and (28). A discount factor is added to the score term so as to prevent the variance of our gradient estimate from growing, again at the cost of an added bias. The convergence analysis of this algorithm requires nonstandard stochastic approximation results developed in [53] and geometric ergodicity results developed in [54].

VI. Summary

In this survey paper, we have discussed nonstandard applications of particle methods to change detection, system identification, and control in nonlinear non-Gaussian state-space models. Many problems which were considered out of reach just a few years ago can now be addressed "routinely" using these simulation-based methods. Other nonstandard applications of particle methods in electrical engineering have been recently developed, including rare event simulation and multitarget tracking using random sets [57]. However, this research area is still very new and many problems remain to be addressed. One of the main issues is reduced variance estimators of the gradients of interest in order to improve the convergence of the stochastic gradient algorithm.

References

[1] C. Andrieu, J. F. G. De Freitas, and A. Doucet, "Sequential MCMC for Bayesian model selection," in *Proc. IEEE Workshop Higher Order Statistics*, 1999, pp. 130–134.

[2] C. Andrieu and A. Doucet, "Online expectation-maximization type algorithms for parameter estimation in general state space models," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 6, 2003, pp. VI-69–VI-72.

[3] C. Andrieu, M. Davy, and A. Doucet, "Efficient particle filtering for jump Markov systems—Applications to time-varying autoregressions," *IEEE Trans. Signal Processing*, vol. 51, pp. 1762–1770, July 2003.

[4] C. Andrieu, A. Doucet, and V. B. Tadíc, "Online sampling for parameter estimation in general state space models," presented at the IFAC Symp. System Identification, Rotterdam, The Netherlands, 2003.

[5] B. Azimi-Sadjadi and P. S. Krishnaprasad, "Change detection for nonlinear systems: A particle filtering approach," presented at the Amer. Control Conf., Anchorage, AK, 2002.

[6] M. Basseville and I. Nikiforov, *Detection of Abrupt Changes—Theory and Application*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[7] A. Benveniste, M. Métivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximation*. New York: Springer-Verlag, 1990.

[8] D. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena, 1995.

[9] F. Cérou, F. LeGland, and N. J. Newton, "Stochastic particle methods for linear tangent equations," in *Optimal Control and PDE's—Innovations and Applications*, J. Menaldi, E. Rofman, and A. Sulem, Eds. Amsterdam, The Netherlands: IOS, 2001, pp. 231–240.

[10] N. Chopin, "A sequential particle filter method for static models," *Biometrika*, vol. 89, pp. 539–552, 2002.

[11] M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks," *Int. J. High Perform. Comput. Appl.*, vol. 6, no. 3, pp. 90–110, 2002.

[12] N. de Freitas, R. Dearden, F. Hutter, R. Morales-Menendez, J. Mutch, and D. Poole, "Diagnosis by a waiter and a Mars explorer," *Proc. IEEE*, vol. 92, pp. 455–468, Mar. 2004.

[13] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-Blackwellised particle filtering for dynamic Bayesian networks," presented at the Conf. Uncertainty Artificial Intelligence, Stanford, CA, 2000.

[14] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Stat. Comput.*, vol. 10, pp. 197–208, 2000.

[15] A. Doucet, J. F. G. de Freitas, and N. J. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.

[16] A. Doucet, N. J. Gordon, and V. K. Krisnamurthy, "Particle filters for state estimation of jump Markov linear systems," *IEEE Trans. Signal Processing*, vol. 49, pp. 613–624, Mar. 2001.

[17] A. Doucet, V. B. Tadić, and S. S. Singh, "Particle methods for average cost control in nonlinear non-Gaussian state space models," in *Proc. Conf. InTech*, Chiang Mai, Thailand, 2003.

[18] A. Doucet, B. Vo, C. Andrieu, and M. Davy, "Particle filtering for multitarget tracking and sensor management," in *Proc. Conf. Fusion*, Annapolis, MD, 2002.

[19] A. Doucet and V. B. Tadić, "Parameter estimation in general state-space models using particle methods," *Ann. Inst. Stat. Math.*, vol. 55, pp. 409–422, 2003.

[20] ——, "Weak derivatives for parameter estimation in general state-space models," Eng. Dept., Cambridge Univ., Tech. Rep., 2003.

[21] R. J. Elliott and V. Krishnamurthy, "New finite dimensional filters for estimation of linear Gauss-Markov models," *IEEE Trans. Automat. Contr.*, vol. 44, pp. 938–951, May 1999.

[22] P. Fearnhead and P. Clifford, "On-line inference for hidden Markov models via particle filters," *J. R. Stat. Soc. B*, vol. 65, pp. 887–899, 2003.

[23] W. R. Gilks and C. Berzuini, "Following a moving target—Monte Carlo inference for dynamic Bayesian models," *J. R. Stat. Soc. B*, vol. 63, pp. 127–146, 2001.

[24] S. J. Godsill, A. Doucet, and M. West, "Monte Carlo smoothing for non-linear time series," Inst. Stat. Decision Sci., Duke Univ., Durham, NC, Tech. Rep. TR00-01, 2000.

[25] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear non-Gaussian Bayesian state estimation," *IEE Proc. F*, vol. 140, pp. 107–113, 1993.

[26] F. Gustaffson, *Adaptive Filtering and Change Detection*. New York: Wiley, 2001.

[27] O. Hernandez-Lerma, *Adaptive Markov Control Processes*. New York: Springer-Verlag, 1989.

[28] T. Higuchi, "Self organizing time series model," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. F. G. de Freitas, and N. J. Gordon, Eds. New York: Springer-Verlag, 2001, pp. 429–444.

[29] M. Hürzeler and H. R. Künsch, "Approximation and maximizing the likelihood for a general state-space model," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. F. G. de Freitas, and N. J. Gordon, Eds. New York: Springer-Verlag, 2001, pp. 159–175.

[30] V. Kadirkamanathan, P. Li, M. H. Jaward, and S. G. Fabri, "Particle filtering-based fault detection in nonlinear stochastic systems," *Int. J. Syst. Sci.*, vol. 33, pp. 259–265, 2002.

[31] K. Kastella, "Discrimination gain for sensor management in multitarget detection and tracking," in *Proc. IEEE-SMC and IMACS Multiconf. CESA*, 1996, pp. 167–172.

[32] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *J. Comput. Graph. Stat.*, vol. 5, pp. 1–25, 1996.

[33] ——, "A self-organizing state-space model," *J. Amer. Stat. Assoc.*, vol. 93, pp. 1203–1215, 1998.

[34] X. Koutsoukos, J. Kurien, and F. Zhao, "Monitoring and diagnosis of hybrid systems using particle filtering methods," in *Proc. 15th Int. Symp. Mathematical Theory Networks and Systems*, Notre Dame, IN, 2002.

[35] F. LeGland and L. Mevel, "Recursive identification in hidden Markov models," in *Proc. 36th IEEE Conf. Decision and Control*, 1997, pp. 3468–3473.

[36] ——, "Fault detection in HMM's: A local asymptotic approach," in *Proc. 39th IEEE Conf. Decision and Control*, 2000, pp. 4686–4690.

[37] U. Lerner, B. Moses, M. Scott, S. McIlraith, and S. Koller, "Monitoring a complex physical system using a hybrid dynamic Bayes net," in *Proc. Conf. Uncertainty in Artificial Intelligence*, Edmonton, AB, Canada, 2002.

[38] P. Li and V. Kadirkamanathan, "Particle filtering based likelihood ratio approach to fault diagnosis in nonlinear stochastic systems," *IEEE Trans. Syst., Man, Cybern. C*, vol. 31, pp. 337–343, Aug. 2001.

[39] J. Liu and M. West, "Combined parameter and state estimation in simulation-based filtering," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. F. G. de Freitas, and N. J. Gordon, Eds. New York: Springer-Verlag, 2001, pp. 197–223.

[40] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*. Cambridge, MA: MIT Press, 1987.

[41] W. S. Lovejoy, "A survey of algorithmic methods for partially observed Markov decision processes," *Ann. Oper. Res.*, vol. 28, pp. 47–66, 1991.

[42] L. Mazliaka, "Approximation of a partially observable stochastic control problem," *Markov Processes Related Fields*, vol. 5, pp. 477–486, 1999.

[43] G. Ch. Pflug, *Optimization of Stochastic Models: The Interface Between Simulation and Optimization*. Boston, MA: Kluwer, 1996.

[44] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filter," *J. Amer. Stat. Assoc.*, vol. 94, pp. 590–599, 1999.

[45] M. K. Pitt, "Smooth likelihood evaluation using particle filters," Department of Economics, Warwick University, Coventry, U.K., Tech. Rep. 651, 2002.

[46] N. G. Polson, J. R. Stroud, and P. Muller, "Pratical filtering with sequential parameter learning," University of Chicago, Chicago, IL, Tech. Rep., 2002.

[47] M. Rosenblatt, "Remarks on a multivariate transformation," *Ann. Math. Stat.*, vol. 23, pp. 470–472, 1952.

[48] T. Rydén, "Consistent and asymptotically normal parameter estimates for hidden Markov models," *Ann. Stat.*, vol. 22, pp. 1884–1895, 1994.

[49] ——, "On recursive estimation for hidden Markov models," *Stoch. Process. Appl.*, vol. 66, pp. 79–96, 1997.

[50] D. Salmond and N. J. Gordon, "Particles and mixtures for tracking and guidance," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. F. G. de Freitas, and N. J. Gordon, Eds. New York: Springer-Verlag, 2001, pp. 517–532.

[51] S. S. Singh, B. Vo, A. Doucet, and R. Evans, "Stochastic approximation for optimal observer trajectory planning," presented at the 42nd IEEE Conf. Decision and Control, Honolulu, HI, 2003.

[52] G. Storvik, "Particle filters in state space models with the presence of unknown static parameters," *IEEE. Trans. Signal Processing*, vol. 50, pp. 281–289, Feb. 2002.

[53] V. B. Tadić, "Asymptotic analysis of stochastic approximation algorithms under violated Kushner-Clark conditions," in *Proc. 39th IEEE Conf. Decision and Control*, vol. 3, 2000, pp. 2875–2880.

[54] V. B. Tadić and A. Doucet, "Exponential forgetting and geometric ergodicity in state space models," in *Proc. 41th IEEE Conf. Decision and Control*, 2002, pp. 2231–2235.

[55] S. Thrun, "Monte Carlo POMDPs," in *Advances in Neural Information Processing Systems*, S. A. Solla, T. K. Leen, and K.-R. Müller, Eds. Cambridge, MA: MIT Press, 2000, vol. 12, pp. 1064–1070.

[56] J. Vermaak, C. Andrieu, A. Doucet, and S. J. Godsill, "Particle methods for Bayesian modeling and enhancement of speech signals," *IEEE Trans. Speech Audio Processing*, vol. 10, pp. 173–185, Mar. 2002.

[57] B. Vo, S. S. Singh, and A. Doucet, "A sequential Monte Carlo implementation of the PHD filter—Applications to multitarget tracking," in *Proc. Conf. Fusion*, Cairns, Qld., Australia, 2003.

[58] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration for tracking applications," in *IEEE Signal Processing Mag.*, vol. 19, Mar. 2002, pp. 61–72.

Christophe Andrieu was born in France in 1968. He received the M.S. degree from the Institut National des Télécommunications, Paris, France, in 1993, and the D.E.A. and Ph.D. degrees from the University of Paris XV, Cergy-Pontoise, France, in 1994 and 1998, respectively.

From 1998 until 2000, he was a Research Associate with the Signal Processing Group, Cambridge University, Cambridge, U.K., and a College Lecturer at Churchill College, Cambridge, U.K. Since 2001, he has been a Lecturer in Statistics at the Department of Mathematics, University of Bristol, Bristol, U.K. His research interests include Bayesian estimation, model selection, Markov chain Monte Carlo methods, sequential Monte Carlo methods (particle filter), stochastic algorithms for optimization with applications to the identification of hidden Markov models, spectral analysis, speech enhancement, source separation, neural networks, communications, and nuclear science, among others.

Arnaud Doucet was born in France on November 2, 1970. He received the M.S. degree from Institut National des Télécommunications, Paris, France, in 1993, and the Ph.D. degree from the University of Paris Sud, Orsay, France, in 1997.

From 1998 to 2001 he was a research associate in Cambridge University, Cambridge, U.K. From 2001 to 2002, he was a Senior Lecturer in the Department of Electrical Engineering of Melbourne University, Melbourne, Australia. Since 2002, he has been a University Lecturer in the Engineering Department, Cambridge University, and a fellow of Churchill College, Cambridge, U.K. He is an associate editor for *The Annals of The Institute of Statistical Mathematics* and *Test*. His research interests include sequential Monte Carlo methods, Markov chain Monte Carlo methods, optimal filtering, and control and reinforcement learning.

Sumeetpal S. Singh received the B.E. (with first-class honors) and Ph.D. degrees from the University of Melbourne, Melbourne, Australia, in 1997 and 2002, respectively.

From 1998 to 1999, he was a Communications Engineer in industry. From 2002 to 2004, he was a Research Fellow in the Department of Electrical and Electronic Engineering, University of Melbourne. He is currently a Research Fellow in the Engineering Department, Cambridge University, Cambridge, U.K. His research interests include the control and optimization of stochastic systems.

Vladislav B. Tadić was born in Belgrade, Serbia and Montenegro, in 1964. He obtained B.Sc., M.Sc. and Ph.D. all in electrical engineering from the University of Belgrade, Serbia and Montenegro, in 1989, 1995, and 1997, respectively. From 1989 to 2000, he held various research and teaching positions at the Mihajlo Pupin Institute, Belgrade, and the Department of Electrical Engineering, University of Belgrade. From 2000 to 2003, he was a Research Fellow in the Department of Electrical and Electronic Engineering, University of Melbourne. Since July 2003, he has been a Lecturer at the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK. His research interests include stochastic approximation, dynamic programming, stochastic optimization, and Monte Carlo methods.